

Dell™ NetVault™ Backup Plug-in for MySQL 4.4

ユーザース・ガイド



© 2014 Dell Inc.
ALL RIGHTS RESERVED.

本ガイドには、著作権法で保護されている機密情報が含まれています。本ガイドに記載されているソフトウェアは、ソフトウェア使用許諾または機密保持契約に基づいて提供されています。本ソフトウェアは、該当する契約書の条件に準拠する限り、使用または複製が許可されます。本ガイドのいかなる部分も、その目的を問わず、複写および記録を含む電子的または機械的な何らかの形式または手段により、Dell Inc. の書面による事前の許可なく、複製または転送することを禁じます。ただし、購入者の個人的な使用については、この限りではありません。

本ドキュメント内の情報は Dell 製品に関連して規定されています。明示あるいは黙示を問わず、禁反言あるいは別の方法で、本ドキュメントから許可を受ける知的所有権あるいは Dell 製品譲渡に関連する知的所有権に対しては、ライセンスはありません。本製品のライセンス契約同様、条件および規約の記載を除いて、DELL は一切の責任を負いません。また、製品に関する黙示的法令保証の権利を放棄します。制限はありませんが、その製品は市場性、特定の目的に対する適用度、または反侵害行為を含む黙示的保証があります。DELL は、損害が生じる可能性について報告を受けたとしても、本ドキュメントの使用、または使用できないことから生じるいかなる、直接的、間接的、必然的、懲罰的、特有または偶発的な障害（無期限、利益の損失、事業中断、情報の損失も含む）に対しても責任を負わないものとします。Dell は、本ドキュメント内容の精密さや完全性について表明および保証しません。また、Quest は告知なしで製品使用や製品解説書を変更する権限があります。Dell は、本ドキュメントに記載されている情報を更新する義務はありません。

本書の使用に関するご質問は、下記までお問い合わせください。

Dell Inc.
Attn:LEGAL Dept
5 Polaris Way
Aliso Viejo, CA 92656

各地域および国際的な当社オフィスの情報については、当社の Web サイト (software.dell.com) をご覧ください。


特許


本製品は米国特許第 7,814,260、7,913,043、7,979,650、8,086,782、8,145,864、8,171,247、8,255,654、8,271,755、8,311,985、8,452,731、および 8,544,023 により保護されています。日本、EU、フランスおよび英国特許第 1615131 および 05250687.0、ドイツ特許 DE602004002858 により保護されています。その他の特許は出願中です。詳しくは、<http://software.dell.com/legal/patents.aspx> を参照してください。

商標

Dell、Dell のロゴ、および NetVault は、Dell Inc. およびその関連会社の商標です。FreeBSD は、The FreeBSD Foundation の登録商標です。Linux は、米国、他国、またはその両方における Linus Torvalds の登録商標です。Mac および OS X は、米国および他国で登録された Apple Inc. の商標です。MySQL は、米国、EU、および他の国々における、MySQL AB の登録商標です。Sun、Oracle、Java、および Oracle Solaris は、米国および他国における Oracle およびその関連会社の商標または登録商標です。SPARC は、米国および他国における SPARC International, Inc. の登録商標です。SPARC の商標を担う製品は、Oracle Corporation が開発したアーキテクチャをベースにしています。Windows は、米国や他国における Microsoft Corporation の登録商標です。このドキュメント内では、その他の商標および商号が、その商標および商号またはその製品の権利を主張する第三者を表すために用いられている場合があります。Dell は、第三者の商標や商号の独占的所有権を否認いたします。

凡例

 **注意：**注意アイコンは、指示に従わなかった場合に、ハードウェアの損傷やデータの損失につながる可能性があることを表しています。

 **警告：**警告アイコンは、潜在的な資産の損害、個人の負傷または死亡の可能性を表しています。

 **重要、メモ、ヒント、モバイル、またはビデオ：**情報アイコンは、サポート情報を表しています。

NetVault Backup Plug-in for MySQL ユーザーズ・ガイド
更新 - 2014 年 4 月
ソフトウェアのバージョン - 4.4
MYG-101-4.4-EN-01

目次

Dell™ NetVault™ Backup Plug-in for MySQL — はじめに	5
Dell NetVault Backup Plug-in for MySQL — 概要	5
主な利点	5
機能概要	6
対象ユーザ	7
参考資料	7
プラグインのインストールと削除	8
インストールの前提条件	8
MySQL サーバでのバイナリ・ログの有効化 (MySQL Standard/Community オプションのみ)	9
推奨構成	10
プラグインのインストール	10
プラグインの削除	11
特定 MySQL インスタンスの削除	11
プラグインの設定	12
デフォルト設定の構成	12
既存インスタンス設定の更新	15
エラー条件のデフォルト・アクションの設定 (オプション)	15
データのバックアップ	17
データのバックアップ — 概要	17
MySQL Standard/Community に関する重要注意事項	17
MySQL Enterprise バックアップに関する重要注意事項	18
バックアップ計画の策定	19
バックアップの実行	22
バックアップ対象データの選択	23
バックアップ・オプションの設定	24
ジョブのファイナライズと実行	26
データのリストア	27
データのリストア — 概要	27
MySQL Standard/Community に利用可能なリストア方法の確認	27
MySQL におけるデータのリストア	28
リストアするデータの選択	28
リストア・オプションの設定	29
ジョブのファイナライズと実行	36
MySQL Standard/Community 用リストア・シナリオ例	36
MySQL Enterprise バックアップ用リストア・シナリオ例	61
高度な MySQL Standard/Community 用リストア手順	62

リストア中にデータベース名を変更する	62
同じサーバ上の別の MySQL インスタンスへリストアする	63
異なる MySQL サーバへのリストア	65
MySQL レプリケーションの使用	68
MySQL レプリケーション環境でのプラグイン使用 – 概要	68
レプリケーションのサポートの有効化	68
レプリケーション・サーバのバックアップ	69
レプリケーション設定のバックアップ	69
レプリケーション・サーバのリストア	69
フェイルオーバー・クラスタ環境での プラグインの使用	71
MySQL サーバ・フェイルオーバー・クラスタリング – 概要	71
重要な注意事項	72
プラグインのインストール	72
インストールの前提条件	72
ソフトウェアのインストール	72
プラグインのライセンス	73
プラグインの設定	73
データのバックアップ	74
データのリストア	74
トラブルシューティング	75
Dell について	76
Dell へのお問い合わせ	76
テクニカル・サポート用リソース	76

Dell™ NetVault™ Backup Plug-in for MySQL — はじめに

- [Dell NetVault Backup Plug-in for MySQL — 概要](#)
- [主な利点](#)
- [機能概要](#)
- [対象ユーザ](#)
- [参考資料](#)

Dell NetVault Backup Plug-in for MySQL — 概要

Dell NetVault Backup (NetVault Backup) Plug-in for MySQL (Plug-in for MySQL) は、複雑なスクリプトを作成せずに、複数の MySQL ストレージ・エンジンのバックアップとリカバリを単一のジョブに統合します。[MySQL Enterprise バックアップ] オプション (MEB ベース・バックアップ方法) を使用する場合、本プラグインはバックアップ時に InnoDB テーブルのホット・バックアップをサポートします。[MySQL Standard/Community] オプション (mysqldump ベース・バックアップ方法) を使用する場合、本プラグインは読み取り専用アクセスでデータをオンラインに保ちながら、すべてのテーブルのウォーム・バックアップをサポートします。また、[MySQL Standard/Community] オプションを使用することにより、Plug-in for MySQL はさらに細かいリストアを実行する高度な特定時点 (PIT) リストア機能を提供します。この機能により正確な時点へのリストアが可能になるため、データの損失を大幅に低減できます。

主な利点

- **MySQL 環境に対する安心感を高め、リスクを低減** — Plug-in for MySQL には、複雑なバックアップ・スクリプトの作成は必要ありません。また、多様なリカバリ・シナリオに対応できる柔軟性を備えています。ユーザは、MySQL に関する詳細な知識がなくても、コミットされたトランザクションがバックアップ中に失われないようにするバックアップ・ポリシーを実装したり、バイナリ・ログをパージする適切なタイミングを把握することができます。この知識はすべてプラグインに組み込まれています。

このほか、Plug-in for MySQL の柔軟なバックアップ機能には以下のものがあります。

- データをオンラインにした状態、すなわちアクセス可能な状態でフル、増分、および差分バックアップを実行
- 複数のストレージ・エンジンに渡って共通のユーザ・インターフェイスを採用
- テーブル・レベルまで保護
- 複数のストレージ・エンジンを単一のジョブに統合

Plug-in for MySQL を使用してバックアップ・ポリシーを実装すると、障害発生時に必要となるリカバリ作業をおろそかにすることなく、より重要なタスクに専念する余裕が生まれます。また、どのような状況であろうと、MySQL データが保護されていることが分かっているため、IT 管理者の安心感が高まります。

- **高速なリストアによりダウンタイムを短縮** – Plug-in for MySQL を使用すれば、リストア・プロセスで必要となる各コマンドを手動で発行したり、リストアを必要とするバイナリ・ログやその適用順序を正確に判断する必要がなくなります。リストアを必要とする対象、リストア元のバックアップ・セット、また該当する場合はリストアする時点または位置を選択するだけで、プラグインが自動的にリストアを実行するため、それ以上の操作は不要です。また、人的な操作への依存を低減することによってリストアを高速化するだけでなく、再起動を必要とする複雑なスクリプトの構文エラーの可能性も排除します。

このほか、Plug-in for MySQL は以下のリストア機能を備えています。

- 時間および位置に基づく特定時点のフルおよび増分リストア
 - インスタンス全体、個々のデータベース、または個々のテーブルのリストア
 - リストア中のデータベース名の変更
 - 代替 MySQL インスタンスへのリストア
- **バックアップ・デバイスの自動統合により業務継続性を確保** – 基幹業務アプリケーションのデータ保護計画においてオフサイト・バックアップが重要な要素となる状況で、Plug-in for MySQL は、Dell NetVault SmartDisk、テープ・ライブラリ、スタンドアロン・テープ・デバイス、仮想テープ・ライブラリ (VTL)、共有 VTL などを含む、幅広いバックアップ・デバイスとの、NetVault Backup の根本的な統合機能を活用しています。NetVault Backup では、バックアップの保存先バックアップ・デバイスを柔軟に選択することができます。バックアップをオンラインで仮想テープに保存したり、複数の MySQL インスタンスやその他の専用データベースで共有される物理テープ・ライブラリ、または一般的なバックアップ用の物理テープ・ライブラリにジョブを複製することができます。
 - **高度な MySQL レプリケーション技術のサポート** – 『MySQL リファレンス・ガイド』で詳しく説明されているように、MySQL は、1 台のサーバがマスタとして、その他の 1 台以上のサーバがスレーブとして機能する、一方向の非同期レプリケーションをサポートしています。

単一マスタ・レプリケーションでは、マスタ・サーバはそのバイナリ・ログに更新を書き込み、これらのファイルのインデックスを管理してログ・ローテーションを追跡します。バイナリ・ログ・ファイルは、スレーブ・サーバに送信される更新の記録として機能します。スレーブはマスタに接続すると、スレーブが最後に正しく更新を行ったときにどこまでログを読み取ったかをマスタに通知します。スレーブは、それ以降に行われた更新を受信してブロックし、マスタが新たな更新を通知するまで待機します。

Plug-in for MySQL は、MySQL 環境が保護され、障害復旧に備えてオフサイトに保存されているという安心感を提供します。同時に、経験の浅い MySQL 管理者でも、リストアが正しく、またできる限り迅速に実行されるという安心感を持ってリストアを開始できるため、ダウンタイムが短縮され、ビジネス継続性が高まり、管理者は常時待機している必要がなくなります。

機能概要

- [MySQL Standard/Community] オプションでは、以下の機能がサポートされます。
 - フル・バックアップと増分バックアップ
 - 差分バックアップ
 - 個々のデータベース / テーブル・コピーのみのバックアップ
 - InnoDB、MyISAM、MERGE (または MRG_MyISAM)、Memory/Heap、Federated、Berkeley DB (BDB)、Archive、および CSV ストレージ・エンジンをサポート
 - 複数のストレージ・エンジンに渡って共通のユーザ・インターフェイスを採用
 - 時間および位置に基づく PIT リストア
 - データ損傷の前および後への PIT リストア
 - 個々のテーブル、データベース、またはインスタンス全体のリストア

- リストア中にデータベース名を変更
- 代替インスタンスへのリストア
- MySQL レプリケーションのスレーブおよびマスタ・インスタンスのバックアップをサポート
- [MySQL Enterprise バックアップ] オプションでは、以下の機能がサポートされます。
 - フル・バックアップと増分バックアップ
 - InnoDB、MyISAM、MERGE（または MRG_MyISAM）、Archive、および CSV ストレージ・エンジン
 - InnoDB テーブルのホット・バックアップ
 - 複数のストレージ・エンジンに渡って共通のユーザ・インターフェイスを採用
 - 個々のテーブル、データベース、またはインスタンス全体のリストア

対象ユーザ

一般的に、ルーチ的なバックアップ・オペレーションの作成および実行に MySQL データベース管理者の高度なスキルは要求されませんが、MySQL バックアップおよびリカバリ戦略を定義したり、高度なリカバリ・シナリオを実行する場合はこのスキルが要求されます。

参考資料

Dell は、本プラグインの設定時および使用中に以下のドキュメンテーションをすぐに利用できるよう準備しておくことをお勧めします。

- 『MySQL <X> リファレンス・マニュアル』（ここで、<X> は、MySQL サーバにインストールされた MySQL バージョンを示します）。
 - MySQL バージョン 5.0 : <http://dev.mysql.com/doc/refman/5.0/en/index.html>
 - MySQL バージョン 5.1 : <http://dev.mysql.com/doc/refman/5.1/en/index.html>
 - MySQL バージョン 5.5 : <http://dev.mysql.com/doc/refman/5.5/en/index.html>
 - MySQL バージョン 5.6 : <http://dev.mysql.com/doc/refman/5.6/en/index.html>
- NetVault Backup ドキュメンテーション
 - 『Dell NetVault Backup インストール・ガイド』— このガイドでは、NetVault Backup サーバおよびクライアント・ソフトウェアのインストール方法について詳しく説明しています。
 - 『Dell NetVault Backup アドミニストレーターズ・ガイド』— このガイドでは、NetVault Backup の使用方法と、すべてのプラグインで共通の機能について詳説します。
 - 『Dell NetVault Backup コマンドライン・インターフェイス・リファレンス・ガイド』— このガイドでは、NetVault Backup のコマンドライン・ユーティリティについて詳しく説明しています。

これらのガイドは、<http://support.software.dell.com/jp/> からダウンロードできます。

- ① **重要：** NetVault Backup は 10.0 から、NetVault Backup システムとインストールされているプラグインを設定、管理、監視するための、Web ベースのユーザ・インターフェイス（WebUI）を提供しています。このバージョンのプラグインのユーザーズ・ガイドに記載されている手順は、この新しい WebUI の使用を前提にしています。NetVault Backup コンソール（NetVault Backup 9.x および 8.x で使用できるユーザ・インターフェイス）による手順について詳しくは、古いバージョンのプラグインのドキュメントを参照してください。

プラグインのインストールと削除

- インストールの前提条件
- 推奨構成
- プラグインのインストール
- プラグインの削除
- 特定 MySQL インスタンスの削除

インストールの前提条件

Plug-in for MySQL のインストールにあたり、MySQL サーバとして使用するマシンに以下のソフトウェアがインストールされ、かつ正しく設定されていることを確認する必要があります。

- **NetVault Backup サーバ/クライアント・ソフトウェア** – MySQL サーバとして構成されたマシン上に、少なくともクライアント・バージョンの NetVault Backup ソフトウェアをインストールする必要があります。
- **MySQL データベース・ソフトウェア**
- **MySQL サーバでバイナリ・ログを有効にする ([MySQL Standard/Community] オプションのみ)** – これにより、MySQL サーバの**特定時点 (PIT) バックアップ**および**リストア**がサポートされます。詳しくは、[MySQL サーバでのバイナリ・ログの有効化 \(MySQL Standard/Community オプションのみ\)](#) を参照してください。
- **適切なバージョンの MySQL データベース・クライアント・パッケージのインストール** – プラグインは、MySQL クライアント・パッケージとともにインストールしたコンポーネントと対話し、プラグインのより多くの機能へのアクセスを可能にします。このパッケージとともにインストールされるコンポーネントのバージョンは、インストールされている MySQL のバージョンと互換性があることが必要です。まず、以下の 2 つの MySQL コンポーネントをインストールし、そのバージョンを確認する必要があります。
 - **mysqldump** – このユーティリティは、さまざまなタイプの MySQL ストレージ・エンジンのバックアップ / リストアを可能にします。このコンポーネントのバージョンが、旧バージョンの Plug-in for MySQL とともに提供されたバージョンではなく、ご利用中の MySQL のバージョンと互換性があることを確認する必要があります。
 - **mysqlbinlog** – このコンポーネントは、データの PIT バックアップおよびリストアを可能にします。このコンポーネントがインストールされている MySQL バージョンと併用できる適切なバージョンであるかを確認する必要があります。
- **[MySQL Enterprise バックアップ]** – **[MySQL Enterprise バックアップ]** オプションを使用する場合、ご使用の環境が以下の要件を満たす必要があります。
 - Linux/UNIX 環境において、MySQL サーバのバージョンが 5.5 または 5.6 であること。
 - Windows 環境において、MySQL サーバのバージョンが 5.5 であること。
 - MySQL Enterprise バックアップ製品のバージョン 3.8.1 がインストールされていること。Commercial Edition を選択した状態の MySQL Enterprise Edition で MySQL Enterprise バックアップが利用可能であること。インストールについて詳しくは、次の URL を参照してください：<http://dev.mysql.com/doc/mysql-enterprise-backup/3.8/en/installing.html>

MySQL サーバでのバイナリ・ログの有効化（MySQL Standard/Community オプションのみ）

[MySQL Standard/Community] オプションを使用した特定時点バックアップ / リストアのサポートを構成する前に、MySQL のバイナリ・ログを有効にする必要があります。

Linux または UNIX ベースの MySQL サーバにおけるログ・オンの有効化

- 1 MySQL インストール・ディレクトリにアクセスし、`my.cnf` という名前の MySQL 構成ファイルを特定します。
このファイルの名前と位置は、ご使用の MySQL 構成によって異なります。詳しくは、MySQL ドキュメンテーションを参照してください。
- 2 テキスト・エディタを使用してファイルを開き、`[mysqld]` セクションに移動します。
- 3 デフォルト MySQL ディレクトリを使用して MySQL バイナリ・ログを有効にするには、以下のエントリを追加します。

`log-bin`

- ① **重要：**バイナリ・ログを有効にするには、以下の構文を使用して `my.cnf` ファイルに追加した `log-bin` エントリを設定することができます：

`log-bin=<宛先ファイル名>`

バイナリ・ログの格納先ファイル名を指定する場合、ファイルへのフルパス情報やファイルの拡張子は含めず、ファイルそのものの名前だけを指定するよう注意してください。プラグインのインストールを実行する前に、上記の手順と共にバイナリ・ログの有効化の詳細について『MySQL リファレンス・マニュアル』を参照してください。

- 4 変更を有効化するには、MySQL サーバを再起動します。

Windows ベースの MySQL サーバにおけるログ・オンの有効化

- 1 MySQL Administrator アプリケーションを起動します（詳しくは MySQL ドキュメンテーションを参照してください）。
 - ① **重要：**MySQL Administrator がまだインストールされていない場合、設定ファイルを更新し、次に MySQL サービスを一旦停止してから再起動してバイナリ・ログを有効化する必要があります。
- 2 [MySQL Administrator] ウィンドウの左ペインで、[Startup Variables] をクリックします。
- 3 右ペインで、[Log Files] タブを選択します。
- 4 [Binary Logfile Name] に、バイナリ・ログ・ファイルの名前として一意の値を入力するか、デフォルト値である `log-bin` を使用するためにフィールドを空白のままにします。

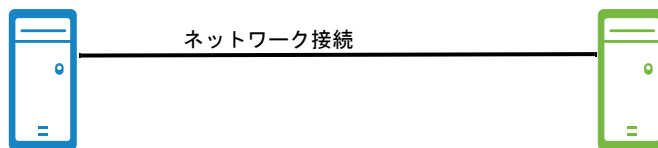
- ① **重要：**バイナリ・ログの格納先ファイル名を指定する場合、ファイルへのフルパス情報やファイルの拡張子は含めず、ファイルそのものの名前だけを指定するよう注意してください。プラグインのインストールを実行する前に、上記の手順と共にバイナリ・ログの有効化の詳細について『MySQL リファレンス・マニュアル』を参照してください。

- 5 MySQL Administrator アプリケーションを閉じます。
- 6 変更を有効化するには、MySQL サーバを再起動します。

推奨構成

1つのマシンを NetVault Backup サーバおよび MySQL サーバの両方として設定する（つまり、すべてのソフトウェアのインストールおよび設定を1つのマシンで実行する）ことはできますが、これらのエンティティは別々のマシンで実行することをお勧めします。

表 1. 推奨構成



MySQL サーバ・マシン	NetVault Backup サーバ・マシン
インストールされているソフトウェア / 構成 <ul style="list-style-type: none">MySQL ソフトウェア（バージョン 5.0 以降）NetVault Backup サーバ / クライアント・ソフトウェアPlug-in for MySQLバイナリ・ログを有効にする（MySQL Standard/Community オプションのみ）mysqldump/mysqlbinlog ユーティリティ（インストールされている MySQL のバージョンとの互換性が必要）mysqlbackup ユーティリティ（[MySQL Enterprise バックアップ] オプションのみ）	インストールされているソフトウェア / 構成 <ul style="list-style-type: none">NetVault Backup サーバ・ソフトウェアNetVault Backup クライアントとして追加された MySQL サーバ（クライアント・マシンを NetVault Backup サーバへ追加する手順について詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください）

❶ **重要：**本書の例の画像および手順では、この2台のマシン環境を使用して、前提条件をすべて満たしていることを前提としています。


プラグインのインストール

プラグインをインストールするには

1 [NetVault 設定ウィザード] または [クライアント管理] ページにアクセスします。


❶ 設定ウィザードを使って、複数のクライアントにプラグインを同時にインストールすることができます（選択するクライアントがすべて同じタイプの場合）。複数のクライアントを選択する場合、プラグインのバイナリ・ファイルがターゲットクライアントの OS とプラットフォームと互換性があることを確認する必要があります。[クライアント管理] ページでは、プラグインをインストールするクライアントを1つのみ選択できます。

- [NetVault 設定ウィザード] ページにアクセスするには：
 - a [ナビゲーション] パネルで、[ガイド付き設定] をクリックします。
 - b [NetVault 設定ウィザード] ページで、[プラグインのインストール] をクリックします。
 - c 次のページで、利用可能なクライアントを選択します。
- [クライアント管理] ページにアクセスするには：
 - a [ナビゲーション] パネルで、[クライアント管理] をクリックします。
 - b [クライアント管理] ページで、利用可能なマシンを選択して、[管理] をクリックします。

- c [クライアント表示] ページで、[プラグインのインストール] ボタン () をクリックします。
- 2 [プラグイン ファイルの選択] をクリックして、プラグイン用 .npk インストール・ファイルの場所 (インストール用 CD または Web サイトからファイルをダウンロードして保存したディレクトリなど) を探します。
インストール CD では、このソフトウェアのディレクトリ・パスはオペレーティング・システムによって異なります。
- 3 ファイル「mys-x-x-x.npk (where xxxx は、プラットフォームおよびそのバージョンを示します) を選択し、[開く] をクリックします。
- 4 [プラグインのインストール] をクリックすると、インストールが開始されます。
プラグインが正常にインストールされると、メッセージが表示されます。

プラグインの削除

プラグインを削除するには

- 1 [ナビゲーション] パネルで、[クライアント管理] をクリックします。
- 2 [クライアント管理] ページで、利用可能なクライアントを選択して、[管理] をクリックします。
- 3 [クライアント表示] ページの [インストール済みソフトウェア] テーブルで、[Plug-in for MySQL] を選択して [プラグインのアンインストール] ボタン () をクリックします。
- 4 [確認] ダイアログで、[削除] をクリックします。

特定 MySQL インスタンスの削除

MySQL インスタンスが正常に構成され、プラグインに追加されたら、これを削除することも可能です。

- ① | **重要:** この手順を実行する際は、注意が必要です。ただし、「[プラグインの設定](#)」で説明する手順に従って、インスタンスを再追加することも可能です。

インスタンスを削除するには

- 1 [ナビゲーション] パネルで、[バックアップ ジョブ作成] をクリックして、次に [セレクション] リストの隣りにある [新規作成] をクリックします。
- 2 セレクション・ツリー内で適切なクライアント・ノードを開きます。
- 3 [Plug-in for MySQL] を開いて、適切なインスタンスを選択します。
- 4 [アクション] リストから、[サーバを削除] を選択します。
ただし、このコマンドを使用しても確認用ダイアログは特に表示されない点に注意してください。

プラグインの設定

- デフォルト設定の構成
- エラー条件のデフォルト・アクションの設定（オプション）

デフォルト設定の構成

Plug-in for MySQL は、単一の MySQL サーバについて複数の MySQL インスタンスをサポートしています。各インスタンスは、用途に応じて設定する必要があります。使用可能な設定オプションは、MySQL サーバで使用されている OS、および **【MySQL Standard/Community】** オプションまたは **【MySQL Enterprise バックアップ】** オプションを使用するかどうかによって異なる点に注意する必要があります。

プラグインを設定するには

- 1 [ナビゲーション] パネルで、**【バックアップ ジョブ作成】** をクリックして、次に **【セレクション】** リストの隣りにある **【新規作成】** をクリックします。
- 2 セレクション・ツリー内で適切なクライアント・ノードを開き、次に **【Plug-in for MySQL】** を選択します。
- 3 **【アクション】** リストから、**【新しいサーバの追加】** を選択します。
- 4 **【設定】** ダイアログで、利用可能なフィールドを記入します。
 - **【MySQL インスタンス名】**（必須）－ MySQL インスタンスにつける名前を入力します（デフォルト値として、ローカル・ホスト名が使用されます）。ここで入力した値は、NetVault Backup WebUI で使用されます。
 - **【MySQL エディション】**（必須）－ **【MySQL Standard/Community】** または **【MySQL Enterprise バックアップ】** を選択します。使用するバージョンにより、このダイアログで変更可能なオプションは異なります。
 - **【ユーザ名】** および **【パスワード】**（必須）－ MySQL インスタンスのデータベース内でバックアップおよびリストアのターゲットとなるテーブルの読み取り / 書き込み権限を持つユーザ（例：管理者権限を持つアカウント）の名前およびパスワードを入力します。
 - ① **重要：** NetVault Backup は、選択されたインスタンスの MySQL データベースにアクセスするたびに、**【ユーザ名】** および **【パスワード】** の値を参照します（バックアップ時およびリストア時の両方）。MySQL インスタンスでこれらの一方の値が変更された場合、これらのフィールドで最新の値に更新する**必要があります**。更新しない場合、NetVault Backup でインスタンスにアクセスできなくなり、ジョブ要求が適切に実行されません。
 - **【MySQL ベース・ディレクトリ】**（必須）－ MySQL インストール・ベース・ディレクトリ（MySQL プログラム・ファイルが保存されているベース・ディレクトリ）のフル・パスを入力します。
 - **Windows ベースの MySQL サーバ**－ MySQL ベース・ディレクトリは、Windows レジストリで「Location」値を照会すると見つかります。
 - **Linux/UNIX ベースの MySQL サーバ**－ MySQL をデフォルト・インストールした場合、MySQL ベース・ディレクトリは以下の場所になります。

```
"/var/lib/mysql"
```

- **[MySQL Bin ディレクトリ]** – MySQL の実行可能ファイルが保存されている、MySQL サーバ上のディレクトリのフル・パスを入力します。デフォルト・インストールした場合、MySQL ベース・ディレクトリは以下の場所になります。
 - Linux/UNIX ベースの MySQL サーバ：


```
"<MySQL ベース・ディレクトリ >/bin"
```
 - Windows ベースの MySQL サーバ：


```
"<MySQL ベース・ディレクトリ >\bin"
```
- **[mysqldump パス]** – バックアップおよびリストア・プロセスで使用される `mysqldump` ユーティリティのフル・パスとファイル名を入力します。ターゲット・インスタンスに対して MySQL のデフォルト・インストールを実行している場合は、MySQL サーバ上で使用されている OS に基づいたファイルへの正しいパスがデフォルト値としてこのフィールドに表示されています。
 - Linux/UNIX ベースの MySQL サーバ：


```
"<MySQL ベース・ディレクトリ >/bin/mysqldump"
```
 - Windows ベースの MySQL サーバ：


```
"<MySQL ベース・ディレクトリ >\bin\mysqldump"
```

① **重要：** [MySQL Bin ディレクトリ] または [mysqldump パス] が上記のデフォルト値に設定されている場合、このフィールドを空白のままにしても構いません。

- **[TCP ポート]** (Windows ベースの MySQL サーバのみ) – 正しくアクセスするために、MySQL の各インスタンスに独自のポート値を指定します。MySQL のデフォルト・インストールではポート **3306** を使用するため、この値がデフォルトで表示されます。選択したインスタンスに対して別のポートを設定した場合は、適切な値を必ず入力する必要があります。

① **重要：** 1 台の MySQL サーバに複数のインスタンスが存在する場合、それぞれに独自のポート値が割り当てられるため、この値を [Port Number] フィールドに入力する必要があります。この値は、各インスタンスの `my.ini` ファイルで「port=」に表示される値と同じです。

- **[ソケット・ファイル・パス]** (Linux/UNIX ベースの MySQL サーバのみ) – MySQL ソケット・ファイルのパスとファイル名を入力します。MySQL のデフォルト・インストールを実行した場合は、正しいパスがデフォルト値としてこのフィールドに表示されます。MySQL をデフォルト・インストールした場合、ソケット・ファイルは以下のディレクトリにあります。

```
"/tmp/mysql.sock"
```

① **重要：** MySQL の標準インストールを MySQL サーバで実行した場合（デフォルトのインストール先ディレクトリを使用した場合は）、[ソケット・ファイル・パス] にはデフォルトのパスをそのまま使用します。ただし、MySQL のインストール時にデフォルト以外のディレクトリを指定した場合は、ソケット・ファイルの正しい場所をこのフィールドに入力する必要があります。このパスを特定するには、MySQL サーバのターミナル・セッションのプロンプトで次のコマンドを実行します。

```
「show variables like 'socket'」
```

[ソケット・ファイル・パス] に正しい値を入力していないと、本プラグインが正常に機能せず、バックアップ/リストアを実行することができません。

- **[デフォルト文字セット]** – デフォルト文字セットには、「latin1」が設定されています。エンコーディングに異なる文字セット (UTF-8 など) を設定したい場合は、リストから選択します。

5 **[MySQL Standard/Community]** を使用している場合は、以下のフィールドを記入します。

- **[MyISAM バックアップ方法]** – ご使用の環境において MyISAM ストレージ・エンジンまたはテーブル・タイプが使用されている場合、利用可能なサブオプションを選択します。

- **[テーブル・ファイルのロック & コピー]** (デフォルト選択) – このオプションを使用してテーブル・ファイルのロック、フラッシュ、およびコピーについて標準バックアップ方法を使用することができます。
 - **[Mysqldump]** – テーブルを多用または頻繁にロードする場合、テーブルをコピーする代わりにこのオプションを選択して **[Mysqldump]** ユーティリティを使用します。ただし、これはパフォーマンスに何らかの影響を及ぼす可能性がある点に注意してください。
 - **[MySQL レプリケーションを可能にする]** – このインスタンスに対してネイティブの MySQL レプリケーションを有効化するには、このチェック・ボックスを選択します。詳しくは、「[MySQL レプリケーションの使用](#)」を参照してください。
 - **[スレーブ・インスタンス]** – MySQL レプリケーションを有効にしたインスタンスについて、そのインスタンスをスレーブ・インスタンスとして設定する場合に選択します。
 - **[マスタ・インスタンス]** – MySQL レプリケーションを有効にしたインスタンスについて、そのインスタンスをマスタ・インスタンスとして設定する場合に選択します。
- ① **重要:** レプリケーションの設定を目的としない場合は、このオプションを選択しないよう注意してください。バックアップが失敗します。
- **[特定時点リカバリを可能にする]** – 特定時点バックアップおよびリストアを有効化した場合、このチェック・ボックスを選択します。名称が示すとおり、この機能を使用すると、データ損傷の前または後、あるいはその両方の特定時点へのリカバリが可能になります。この機能を使用するには、MySQL バイナリ・ログを有効化する必要があります。詳しくは、[MySQL サーバでのバイナリ・ログの有効化 \(MySQL Standard/Community オプションのみ\)](#) を参照してください。
 - **[バイナリ・ログ・インデックス・パス]** – **[特定時点リカバリを可能にする]** チェックボックスを選択した場合、このフィールドを使用してバイナリ・ログ・インデックス・ファイルへのフル・パスを指定します。MySQL をデフォルト・インストールした場合、このファイルは以下の場所にあります。
 - **Linux/UNIX サーバ:**
`<MySQL ベース・ディレクトリ >/data/<インスタンス名 >-bin.index`
 - **Windows サーバ:**
`<MySQL ベース・ディレクトリ >\data\<インスタンス名 >-bin.index`
- ① **重要:** 本プラグインは、**[バイナリ・ログ・インデックス・パス]** に指定されたファイルが存在するかどうかを設定中に判断します。ただし、指定されたファイルが実際にバイナリ・ログ・インデックスかどうかは、バックアップ・ジョブを開始するまで判断できません。指定されたファイル名が有効なバイナリ・ログ・インデックスではないことが判明した場合、ジョブは適切に実行されないことに注意してください。
- **[リレー・ログ・インデックス・パス]** – **[スレーブ・インスタンス]** を設定中の場合、このフィールドを利用してリレー・ログ・インデックス・ファイルへのフル・パスを入力し、バックアップに含めます。
 - **Linux/UNIX サーバ:**
`<MySQL ベース・ディレクトリ >/data/<インスタンス名 >-relay-bin.index`
 - **Windows サーバ:**
`<MySQL ベース・ディレクトリ >\data\<インスタンス名 >-relay-bin.index`
- 6 **[MySQL Enterprise バックアップ]** を使用している場合は、必須の **[Mysqbackup Path]** フィールドに、mysqbackup ユーティリティが存在しているディレクトリへの完全パスを入力します。
- 7 **[OK]** をクリックして設定を保存します。

既存インスタンス設定の更新

MySQL のインスタンスを正しく設定し、プラグインに追加したら、以下の手順に従って設定オプションを編集することができます。

- 1 [ナビゲーション] パネルで、[バックアップ ジョブ作成] をクリックして、次に [セクション] リストの隣りにある [新規作成] をクリックします。
- 2 セクション・ツリー内で適切なクライアント・ノードを開きます。
- 3 [Plug-in for MySQL] ノードを開いて、適切なインスタンスを選択します。
- 4 [アクション] リストから、[設定] を選択します。

[設定] ウィンドウに以前のすべての設定が表示されます。必要に応じてこれらを修正することができます。

- ❶ **重要：** [設定] ウィンドウの編集画面では、[MySQL インスタンス名] フィールドはグレー表示され修正することはできません。これは情報を表示する目的のみのフィールドであり、現在選択されている MySQL インスタンス名が表示されます。

エラー条件のデフォルト・アクションの設定 (オプション)

通常、MySQL バックアップ・ジョブでは、複数のストレージ・エンジン、データベース、テーブルが使用されます。場合によっては、バックアップ・ジョブの実行中に、サポートされていないストレージ・エンジンが検出されたりデータベース / テーブルにアクセスできないために、一部のアイテムが正しくバックアップされない一方で、バックアップ・ジョブで選択された残りのアイテムについては問題なくバックアップされることがあります。MySQL データベース管理者は、このような状況が発生したときにどのようなアクションを実行するかを決定する必要があります。

- バックアップ・ジョブを警告付きで完了するか、警告なしで完了するか、または失敗とするか
- 正しく完了しなかったアイテムのバックアップを保持するか、削除するか

プラグインでは、バックアップおよびリストア・ジョブのデフォルト・オプションを設定できます。これらのオプションは、ジョブごとに上書きできます。

デフォルトを設定するには

- 1 [ナビゲーション] パネルで、[設定変更] をクリックします。
- 2 [設定] ページで、利用可能な [サーバー設定] または [クライアント設定] をクリックします。
- 3 [クライアント設定] を選択した場合は、適切なクライアントを選択して、[次へ] をクリックします。
- 4 [NetVault サーバ設定] または [クライアント設定] ページで、[プラグイン・オプション] をクリックします。

[Plug-in for MySQL] セクションには、以下の項目が記載されています。

- **[ロックされたテーブル]** — この条件は、バックアップの対象として選択されているテーブルがプラグイン以外のクライアント・セッションによってロックされているために、プラグインがテーブルをバックアップできない場合に満たされます。
- **[手動で選択されたテーブルが使用不能]** — この条件は、バックアップの対象として手動で選択または明示的にクリックした個々のテーブルを、バックアップ・ジョブの定義以降に削除されたなど何らかの理由でバックアップできない場合に満たされます。
- **[手動で選択されたデータベースが使用不能]** — この条件は、バックアップの対象として手動で選択または明示的にクリックした個々のデータベースを、バックアップ・ジョブの定義以降に削除されたなど何らかの理由でバックアップできない場合に満たされます。

- **【サポートされていないストレージエンジン】** – この条件は、バックアップ中に、現在プラグインでサポートされていないストレージ・エンジン・タイプのテーブルが検出された場合に満たされます。
- 5 これらの各オプションに対し、以下の4つの設定からいずれかを選択できます。
- **【Complete with Warning - Saveset Retained】** – ジョブが**【警告で完了】**というステータスを返し、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - **【Complete without Warnings - Saveset Retained】** – ジョブが完了し、**【ジョブ終了】**というステータスを返します。エラーはNetVault Backup バイナリ・ログに記録されますが、基本的に、**【ジョブ・ステータス】** ページでは無視され、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - **【Fail - Saveset Retained】** – バックアップ・ジョブは、**【バックアップ・ジョブ失敗】**というステータスを返しますが、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - **【Fail - No Saveset Retained】** – バックアップ・ジョブは、**【バックアップ・ジョブ失敗】**というステータスを返し、バックアップされたオブジェクトのセーブセットは保持されません。選択された一部のオブジェクトが正しくバックアップされた場合も削除されます。
- ① **重要：** 個別のバックアップ・ジョブ・レベルで選択したデフォルト・アクションを選択して上書きすることが可能です。
- 6 **【適用】** をクリックして設定を保存します。

データのバックアップ

- データのバックアップ – 概要
- バックアップの実行

データのバックアップ – 概要

バックアップを完了する前に、以下のセクションの情報を確認します。

- [MySQL Standard/Community に関する重要注意事項](#)
- [MySQL Enterprise バックアップに関する重要注意事項](#)
- [バックアップ計画の策定](#)

MySQL Standard/Community に関する重要注意事項

ハイフンなどの特殊文字を含むデータベースが環境内に存在する場合（半角英数字およびアンダースコアを除くその他すべての文字は、特殊文字として扱われます）、以下の制限事項に注意する必要があります。

- データベース名にハイフンが含まれていても、MyISAM を除くすべてのテーブル・タイプがバックアップされます。これは、mysqldump コマンドが常に上記のテーブル・タイプに使用されるためです。
- データベース名にハイフンが含まれると、**MyISAM バックアップ方法**がバージョン 4.2 で導入された **Mysqldump** オプションに設定されている場合、MyISAM テーブルがバックアップされます。この場合、バックアップおよびリストアのパフォーマンスにマイナスとなる影響を及ぼす場合がある点に注意してください。
- **MyISAM バックアップ方法**にデフォルトの **【テーブル・ファイルのロック & コピー】** オプションを使用し、データベース名にハイフンが使用されていると、MyISAM テーブルはバックアップされません。これは、本プラグインが MySQL コマンドを迂回し、直接テーブル・ファイルのコピーを試行するためです。本プラグインは、テーブル・ファイルが特定できないことを示すエラー・メッセージを出力し、バックアップ・ジョブがセーブセットを作成することなく失敗します。

以前のバージョンでは、本プラグインはデータベース・ディレクトリが存在するか検証を試み、失敗すると警告メッセージを出力してから次のデータベースのバックアップを継続しようとします。この場合、バックアップは警告付きで完了し、その他すべてのデータベースを含むセーブセットを作成します。

Mysqldump オプションを使用したパフォーマンスが最適でないなど、何らかの理由でオリジナルの動作を維持して **【テーブル・ファイルのロック & コピー】** オプションを使用する場合、本プラグイン設定ファイル「**nvmysql.cfg**」の **ValidateDatabaseDirectory** パラメータを以下のように手動で「TRUE」に設定することができます。

```
[MySql:ValidateDatabaseDirectory]
Value=TRUE
```

その後、新規動作を適用しようと決めた場合、「**nvmysql.cfg**」ファイルでこのパラメータを「FALSE」に変更するか削除することができます。

MySQL Enterprise バックアップに関する重要注意事項

[MySQL Enterprise バックアップ] オプションを使用する場合、以下のガイドラインと情報について確認する必要があります。

- MySQL は、重要なデータには InnoDB テーブルを使用することを推奨しています。これは、バックアップ・プロセスが高速で、信頼性および拡張性が高いためです。MySQL Enterprise バックアップは MySQL テーブルについてさまざまなバックアップを提供し、InnoDB テーブルに関するバックアップを最適化します。このオプションは、すべての InnoDB テーブルのホット・バックアップを実行します。データベースの実行中においてもホット・バックアップが実行されるため、バックアップ中に現行データベース操作を停止する必要はありません。さらに、バックアップ・プロセス中にデータベースに加わった変更が含まれます。環境内のデータベースをオンラインに保ち、その拡張性もサポートされている必要がある場合、バックアップの完了に必要な時間に影響を及ぼします。
- このオプションを使用すると、MyISAM テーブルとその他の非 InnoDB テーブルは、ウォーム・バックアップを使用して、最後にバックアップされます。ウォーム・バックアップにおいて、データベースは稼働し続けますが、バックアップが完了してもテーブルは読み込み専用アクセスに設定されます。
- ホット・バックアップ・フェーズにおいてデータの大半がバックアップされたか確認したら、InnoDB を新規テーブル用デフォルト・ストレージ・エンジンとし、既存のテーブルを変換して InnoDB ストレージ・エンジンで利用可能にすることを検討するようお勧めします (MySQL Server 5.5 以降では、InnoDB がデフォルトです)。
- 増分バックアップは主に InnoDB テーブル用であり、非 InnoDB テーブルは読み取り専用、あるいは、それほど頻繁な更新を必要としない用途に向いています。InnoDB ファイルについて、最後のバックアップ以降に変更が起こった場合、ファイル全体が含まれます。
- 本プラグインを使用する場合、以下の条件に適合すると、MySQL インスタンス内のすべての InnoDB テーブルがバックアップされます。

- ただし、テーブルがバックアップ用に明示的に選択され、テーブルが InnoDB ストレージ・エンジンまたはその類ではない場合に限りません。

例：2つのデータベースを含む MySQL インスタンスが配置されています (DB1 および DB2)。各データベースには2つのテーブルが含まれます。DB1 には T1_InnoDB と T1_MyISAM、DB2 には T2_InnoDB と T2_MyISAM が含まれます。T1_MyISAM と T2_MyISAM をバックアップすると、T1_InnoDB と T2_InnoDB もバックアップに含まれます。InnoDB テーブルの1つを含めると、InnoDB テーブルのみがバックアップされます。データベースの1つを選択すると、データベース内のテーブルのみがバックアップされます。

- いくつかまたはすべてのデータベースがバックアップに選択されると、すべての関連 InnoDB テーブルがバックアップから除外されます。

例：2つのデータベースを含む MySQL インスタンスが配置されています (DB1 および DB2)。各データベースには2つのテーブルが含まれます。DB1 には T1_InnoDB と T1_MyISAM、DB2 には T2_InnoDB と T2_MyISAM が含まれます。DB1 と DB2 をバックアップし、T1_InnoDB と T2_InnoDB を除外すると、T1_InnoDB と T2_InnoDB もバックアップに含まれます。2つの InnoDB テーブルのうち1つだけを除外すると、InnoDB テーブルのみがバックアップされます。

これは、MySQL Enterprise バックアップの現在の動作ですが (mysqlbackup ユーティリティ)、MySQL の将来のリリース (3.8.1 以降) では変更される可能性があります。

- MySQL 5.5 では、`innodb_file_per_table` 設定オプションはデフォルトで無効化されています。MySQL 5.6 では、この設定オプションはデフォルトで有効化されています。`innodb_file_per_table` オプションが無効化された状態で作成されたすべての InnoDB テーブルは、InnoDB システム・テーブルスペース内に格納されますが、バックアップから除外することはできません。InnoDB テーブルをテーブルスペース外に配置する必要がある場合、`innodb_file_per_table` オプションを有効化した状態で MySQL 内で InnoDB テーブルを作成する必要があります。各 .ibd ファイルには、1つのテーブルのデータとインデックスのみが含まれます。

バックアップ計画の策定

MySQL バックアップ戦略を定義する際、以下の点を明確にする必要があります。

- **【MySQL Standard/Community】** オプションまたは **【MySQL Enterprise バックアップ】** オプションのどちらを使用するか環境に両方のバージョンを導入している場合でも、プラグインでは1つの計画のみを使用することができます。MEB ベース方法または mysqldump ベース方法のいずれかを使用する必要がありますが、両方を併用することはできない点に注意してください。

MEB ベース・オプションを使用すると、バックアップに選択したすべてのデータベース・オブジェクトに対して `mysqlbackup` コマンドが1度実行され、ジョブ・ログ内に `mysqlbackup` ログが出力されます。データのバックアップには2つのステージが含まれます。最初のステージでは、すべての InnoDB テーブルがコピーされます。2番目のステージでは、すべてのテーブル・タイプがコピーされます。InnoDB テーブルのホット・バックアップをサポートするだけでなく、MEB ベース・オプションはバックアップ・パフォーマンスを向上させます。

mysqldump ベース・オプションを使用する場合、各テーブル、トリガ、およびストアド・プロシージャに対してコマンドが実行されます（ホット・バックアップはサポートされません）。

- フル・バックアップ時にインスタンス全体の読取り専用アクセスが必要なことを認識した上で、フル・バックアップをどのような頻度で実行するか
- バックアップの速度とリストアの速度のどちらを重視するか
- 許容できる最大データ損失量

上記の点を明確にしておくこと、実装するバックアップ・タイプおよび頻度を定義する際に役立ちます。

- [MySQL Standard/Community 用バックアップ・タイプの確認](#)
- [MySQL Enterprise バックアップ用バックアップ・タイプの確認](#)
- [MySQL Standard/Community 用バックアップ・シーケンス例](#)

MySQL Standard/Community 用バックアップ・タイプの確認

【MySQL Standard/Community】 オプションを使用する場合、本プラグインは `mysqldump` を使用して以下のタイプのバックアップを実行します。

- **フル・バックアップ**
- **増分バックアップ**
- **差分バックアップ**
- **個々のデータベース / テーブル・コピーのみのバックアップ**

各 MySQL インスタンスのデータ保護要件に適したバックアップ・シーケンスを選択するには、まずこれらのバックアップの違いを理解する必要があります。

フル・バックアップ

【MySQL Standard/Community】 オプション用フル・バックアップの場合、本プラグインは `mysqldump` ユーティリティを使用して、インスタンスに含まれるすべてのデータベースをバックアップします。フル・バックアップはほぼすべてのリストア・シナリオの起点になるため、あらゆるバックアップ戦略の基盤となります。プラグインで生成されたフル・バックアップを使用して、インスタンス全体、個々または複数のデータベース、個々または複数のテーブルをリストアできます。

【MySQL レプリケーションを可能にする】 オプションが無効化され、**【特定時点リカバリを可能にする】** オプションが有効化されている場合、**【フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ】** オプションはデフォルトで有効化されますが、フルまたは増分バックアップ後に MySQL バイナリ・ログをパージする必要があります。

[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ] オプションを選択しない場合、プラグインは設定ファイルで**最終バックアップのログ**を追跡し続けます。これにより必要に応じてバイナリ・ログを手動でパージすることができます。例えば、MySQL レプリケーション環境で、バイナリ・ログがスレーブ・インスタンスにレプリケートされるまで、バイナリ・ログをマスター・インスタンスからパージしたくない場合などは、ユーザはバイナリ・ログを手動でパージする必要があります。

増分バックアップ

増分バックアップでは、最後のフルまたは増分バックアップ以降に生成されたバイナリ（トランザクション）・ログをバックアップし、次にバイナリ・ログをパージします。バイナリ・ログはインスタンスに基づくため、すべてのデータベースのトランザクション・ログがまとめてバックアップされ、パージされます。

増分バックアップを使用すると、不正な更新やテーブルの削除などのデータ損傷の前および後の時点にリストアできるため、メディア障害またはデータ損傷の発生後のデータ損失を低減する上で重要です。フル・バックアップとは異なり、増分バックアップはバックアップ中に読み取り専用アクセスを必要としません。

MySQL の増分バックアップを実行するには、バイナリ・ログを有効にする `-log-bin` オプションを使用して MySQL インスタンスを開始する必要があります。この手順は、[MySQL サーバでのバイナリ・ログの有効化（MySQL Standard/Community オプションのみ）](#)で概説しています。詳しくは、『MySQL リファレンス・マニュアル』のバイナリ・ログに関するセクションを参照してください。

上記で説明したように、[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ] オプションを使用して、フルまたはインクリメンタル・バックアップ後にバイナリ・ログをパージする必要があります。このオプションを使用しない場合は、本プラグインは設定ファイルで**最終バックアップのログ**を追跡し続けます。これにより必要に応じてバイナリ・ログを手動でパージすることができます。

差分バックアップ

差分バックアップでは、最後のフルまたは増分バックアップ以降に生成されたバイナリ（トランザクション）・ログをバックアップします。ただし、この形式のバックアップでは、完了時にバイナリ・ログが**パージされません**。このため、各差分バックアップには、前の差分バックアップにも含まれていたバイナリ・ログおよび前の差分バックアップ以降に生成されたバイナリ・ログが含まれることになり、以降の差分バックアップのサイズが大きくなるだけでなく、その時間も長くなります。例えば、月曜日から金曜日までの差分バックアップを伴って、日曜日にフル・バックアップの実行がスケジュールされている場合、月曜日の差分には日曜日のフル・バックアップ以降生成されたトランザクション・ログ・ファイルが含まれます。一方、火曜日の差分には、月曜日に生成されたバイナリ・ログ・ファイルだけでなく、火曜日に生成されたバイナリ・ログ・ファイルも含まれます。水曜日の差分バックアップには、月曜日、火曜日、および水曜日のバイナリ・ログが含まれる、というようになります。

増分バックアップと同様に、差分バックアップを使用すると、メディア障害またはデータ損傷が発生した場合のデータ損失を低減でき、障害 / 損傷の前および後の時点にリストアできます。フル・バックアップとは異なり、差分バックアップはバックアップ中に読み取り専用アクセスを必要としません。

差分バックアップを実行するには、バイナリ・ログを有効にする `-log-bin` オプションを使用して MySQL インスタンスを開始する必要があります。この手順は、[MySQL サーバでのバイナリ・ログの有効化（MySQL Standard/Community オプションのみ）](#)で概説しています。詳しくは、『MySQL リファレンス・マニュアル』のバイナリ・ログに関するセクションを参照してください。

増分バックアップと差分バックアップの比較

増分バックアップでは、バイナリ・ログがバックアップ後にパージされ、最後の増分バックアップ後に作成されたバイナリ・ログのみがバックアップされるため、以降の増分バックアップの実行時間は短くなります。ただし、増分バックアップを使用するリストア・シーケンスでは、フル・バックアップから障害時点までに実行されたすべての増分バックアップを継続してリストアする必要があります。このため、複数のリストア・ジョブを開始するためにデータベース管理者に必要な操作が多くなり、リストアに長い時間がかかる可能性があります。

差分バックアップでは、バイナリ・ログがバックアップ後にパージされず、最後のフル・バックアップ後に作成されたすべてのバイナリ・ログがバックアップの対象となるため、以降の各差分バックアップの実行時間は長くなります。ただし、差分バックアップを使用するリストア・シーケンスでは、フル・バックアップのリストア後に差分バックアップを1つのみリストアするだけで済みます。このため、リストア・プロセスでデータベース管理者に必要な操作が少なくなり、リストア時間は短くなります。

個々のデータベース / テーブル・コピーのみのバックアップ

場合によっては、データベース全体の包括的なバックアップおよびリストア手順に影響を与えることなく、特殊な目的でバックアップを実行しなければならないことがあります。たとえば、バックアップをテスト環境のソースにしたり、レプリケーション・スレーブ・インスタンスの初期同期用に使用する場合などです。個々のデータベース / テーブル・コピーのみのバックアップは、このような特殊な目的のために設計されており、MySQL 環境をコピーすることができます。コピーのみのバックアップは、設定されたバックアップ・シーケンスから独立しているため、フル、増分、または差分バックアップのリカバリ可能性には影響しません。ただし、フル・バックアップの代わりとして使用することはできません。

MySQL Enterprise バックアップ用バックアップ・タイプの確認

[MySQL Enterprise バックアップ] オプションについて、本プラグインは選択されたすべてのデータベース・オブジェクトに対して `mysqlbackup` コマンドを1度実行し、以下のタイプのバックアップをアーカイブします。フルと増分。

フル・バックアップ

[MySQL Enterprise バックアップ] オプション用フル・バックアップの場合、本プラグインは `mysqldump` ユーティリティを使用して、インスタンスに含まれるすべてのデータベース・オブジェクトをバックアップします。フル・バックアップはほぼすべてのリストア・シナリオの起点になるため、あらゆるバックアップ戦略の基盤となります。プラグインで生成されたフル・バックアップを使用して、インスタンス全体、個々または複数のデータベース、個々または複数のテーブルをリストアできます。

増分バックアップ

InnoDB テーブルについて、最後のフルまたは増分バックアップ以降に変更が加わったデータのみがバックアップされます。非 InnoDB テーブルの場合、最後のフルまたは増分バックアップ以降に何かテーブル内で変更された場合、テーブル全体がバックアップされます。

MySQL Standard/Community 用バックアップ・シーケンス例

以下のセクションで、さまざまなバックアップ・シーケンスを示します。

フル・バックアップのみ

ビジネス要件で前日までのデータ保護が保証されており、読み取り専用アクセスが毎日許容される場合（勤務時間外など）、フル・バックアップのみを毎日実行すれば十分でしょう。データベース管理者は、データベースを最後のフル・バックアップ時点にリカバリすることはできませんが、MySQL サーバに存在しているバイナリ・ログを使用して PIT リカバリを実行することはできません。

フル・バックアップと増分バックアップ

ビジネス要件で前日までのデータ保護が保証されているが、ターゲット MySQL インスタンスへの読み取り専用アクセスが断続的にのみ許容され（週 1 回または隔週の勤務時間外など）、バックアップ時間をできる限り短縮する必要がある場合、フル・バックアップと増分バックアップの組合せが最適です。たとえば、毎週日曜日の夜 11:00 にフル・バックアップが実行され、さらに月曜日から土曜日の午後 11:00 にトランザクション・ログ・ファイルのバックアップが実行されているとします。この場合、各増分バックアップには、前夜のバックアップ、すなわち日曜日の夜に実行されたフル・バックアップまたはいずれかの増分バックアップ以降に生成されたバイナリ・ログが含まれます。

このバックアップ・タイプ・シーケンスのリストアには、より長い時間がかかることに注意してください。たとえば、火曜日にリカバリを実行する場合、日曜日のフル・バックアップと月曜日の増分バックアップをリストアする必要があります。また、木曜日にリカバリを実行する場合は、日曜日のフル・バックアップに続いて、月曜日、火曜日、および水曜日の増分バックアップをリストアする必要があります。バックアップ時間は短くなりますが、複数のリストア・ジョブを実行するために必要な操作が多くなるため、リストア時間は長くなる可能性があります。

フル・バックアップと差分バックアップ

ビジネス要件で前日までのデータ保護が保証されているが、ターゲット MySQL インスタンスへの読み取り専用アクセスが断続的にのみ許容され（週 1 回または隔週の勤務時間外など）、リストア時間をできる限り短縮する必要がある場合、フル・バックアップと差分バックアップの組合せが最適です。たとえば、フル・バックアップを毎週日曜日の夜 11:00 に実行し、差分バックアップを月曜日から土曜日の午後 11:00 に実行します。各差分バックアップには、最後のフル・バックアップ以降に生成されたバイナリ・ログが含まれます。前述のように、この場合の全体的なバックアップ時間は長くなります。

リカバリする必要がある特定時点に関わらず、必要なリストア・ジョブの数は同じです。たとえば、火曜日にリカバリを実行する場合、日曜日のフル・バックアップと月曜日の差分バックアップをリストアする必要があります。また、木曜日にリカバリを実行する場合は、日曜日のフル・バックアップに続いて水曜日の差分バックアップをリストアする必要があります。以降の差分バックアップはサイズが大きくなり、時間も長くなりますが、実行する必要があるリストア・ジョブの数は少なくなるため、リストア時間は短くなります。

バックアップの実行

Plug-in for MySQL を使用してバックアップを実行するには、以下のセクションで説明する手順に従います。

- [バックアップ対象データの選択](#)
- [バックアップ・オプションの設定](#)
- [ジョブのファイナライズと実行](#)

バックアップ対象データの選択

バックアップ・ジョブを作成するには、セット（バックアップ・セレクション・セット、バックアップ・オプション・セット、スケジュール・セット、ターゲット・セット、および詳細設定セット）を使用する必要があります。

バックアップ・セレクション・セットは、増分および差分バックアップに必要です。フル・バックアップを実行中には、バックアップ・セレクション・セットを作成してからフル、増分、差分バックアップに使用する必要があります。増分または差分バックアップにセレクション・セットが使用されていない場合、バックアップ・ジョブがエラーをレポートします。詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

バックアップ・セレクション・セットを作成するには

- 1 [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックします。

[ガイド付き設定] リンクからウィザードを開始することもできます。[ナビゲーション] パネルで、[ガイド付き設定] をクリックします。[NetVault 設定ウィザード] ページで、[バックアップ・ジョブ作成] をクリックします。

- 2 [ジョブ名] に、ジョブの名前を指定します。

ジョブの進捗状況の監視やデータのリストアップ時にジョブを識別しやすくするため、分かりやすい名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できませんが、英語以外の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

- 3 [セレクション] リストの隣りにある [新規作成] をクリックします。

- 4 [NetVault Backup セレクション] ページの [バックアップ・セレクション・セット] ボックスにセットの名前を入力して、次にプラグインがインストールされている NetVault Backup クライアントを開きます。

- 5 プラグインのリストで [Plug-in for MySQL] を開くと、追加し MySQL サーバが表示されます。

- 6 利用可能データを選択します。

- バックアップ・ジョブ内で選択したインスタンスからすべての MySQL データベースを選択するには、[全てのデータベース] ノードを選択します。
- さらに細かく選択するには、[全てのデータベース] ノード内のデータベースを個別に表示させます。さらに、個々のデータベースを開き、その中のテーブルを表示して個別に選択し、バックアップ・ジョブに追加することもできます。
- アイテムをバックアップから明示的に除外するには、親レベルのアイテムを選択し、子のアイテムを個別にクリックして、緑のチェックマークを赤色の X に変更し、除外対象に指定します。

- ① **重要：** [MySQL Standard/Community] オプションを使用したバックアップに細かいレベルのデータ・セットを選択する場合、[バックアップオプション] タブでバックアップ・タイプに [個々のデータベース/テーブル・コピーのみ] を選択する必要があります。他の形式のバックアップ（フル、増分、または差分バックアップ）を選択した場合、細かいレベルの選択は無視されデータベース全体がバックアップされます。MySQL 5.0.x 以降の場合、ストアード・プロシージャ、関数、およびトリガは、[MySQL Standard/Community] オプションの [フル] および [個々のデータベース/テーブル・コピーのみ] で自動的にバックアップされます。

MySQL 5.0.x（以降）では、選択ツリーに information_schema データベースが表示されますが、このデータベースは選択できません。これは、このデータベースに含まれるすべてのデータが動的に生成され、永久的に存在するものではないからです。このため、プラグインは、すべてのバックアップから information_schema データベースを自動的に除外します。

- 7 [保存] をクリックして、セットを保存します。

既存のバックアップ・セレクション・セットを使用するには

- 1 [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックします。

[ガイド付き設定] リンクからウィザードを開始することもできます。[ナビゲーション] パネルで、[ガイド付き設定] をクリックします。[NetVault 設定ウィザード] ページで、[バックアップ・ジョブ作成] をクリックします。

- 2 [選択] リストで、既存のバックアップ・セレクション・セットを選択します。

バックアップ・オプションの設定

次の手順には、バックアップ・オプション・セットの作成または既存のセットの選択が含まれています。[MySQL Standard/Community] または [MySQL Enterprise バックアップ] オプションのいずれかを使用するかによって、[バックアップオプション] で各設定が利用可能になります。

MySQL Standard/Community 用バックアップ・オプションの設定

バックアップする目的のアイテムを選択した状態で、実行するバックアップ・タイプを選択したり、失敗した場合に異なる動作を選択することも出来ます。

バックアップ・オプション・セットを作成するには

- 1 [プラグイン・オプション] リストの隣りにある [新規作成] をクリックします。
- 2 利用可能なオプションを選択します。

❶ **重要:** MySQL のターゲット・インスタンスとしてレプリケーションのマスタ・インスタンスを指定した場合 (この MySQL インスタンスの [設定] ウィンドウで、[MySQL レプリケーションを可能にする] オプションと [マスタ・インスタンス] オプションが選択されている状態)、フル、増分および差分形式のバックアップは選択することはできません。詳しくは、「[MySQL レプリケーションの使用](#)」を参照してください。

- [全てのデータベースをフルバックアップ] (デフォルト選択) — 現在の MySQL インスタンスに含まれるすべてのデータベースについて完全なフル・バックアップを実行するには、このオプションを選択します。
- [インクリメンタル・バックアップ] — 最後のフルまたは増分バックアップ以降に生成されたトランザクション・ログのみをバックアップする場合、このオプションを選択します。
- [ディファレンシャル・バックアップ] — 最後のフルまたは増分バックアップ以降に生成されたすべてのトランザクション・ログをバックアップする場合、このオプションを選択します。以降の差分バックアップには、元のフル・バックアップの実行後に生成されたすべてのバイナリ・ログ・ファイルが含まれます。バックアップ完了後は、MySQL インスタンスのバイナリ・ログは保持されます。
- [個々のデータベース/テーブル・コピーのみ] — 特殊な目的 (テスト環境を作成する場合など) で MySQL 環境をコピーする場合に、このオプションを選択します。データベースの全体的なバックアップおよびリストア手順には影響を与えません。この方法で作成したコピー・バックアップは、MySQL のフル・バックアップと増分 / 差分バックアップを併用したシナリオで設定されたシーケンスに影響しません (これらのバックアップは、バイナリ・ログの記録に影響を与えません)。この形式のバックアップは、MySQL のフル・バックアップと増分 / 差分バックアップの併用シナリオで通常のバックアップ・シーケンスとして設定されたバックアップから独立しています。また、コピー・バックアップをフル・バックアップの代わりに使用することはできません。
- [バックアップ中は、全てのテーブルを読み取りアクセスにロックしトランザクションの消失を防ぎます。] — [フル・バックアップ] を選択し、現在読み取り専用アクセスのインスタンス内のすべてのデータベースがロックされることによるトランザクションの消失を防ぎたい場合に、このオプションを選択します。このオプションを選択した場合、ユーザはフル・バックアップ中にイ

インスタンス全体でデータの挿入、更新、または削除を実行できません。このオプションが選択解除されると、Plug-in for MySQL は、テーブルがバックアップされた場合に限りバックアップ・プロセス中に各テーブルをロックします。このため、インスタンスに関連するテーブルが含まれる場合、このオプションを選択してバックアップ・プロセス中に確実にすべてのテーブルがロックされるようにすることをお勧めします。

- [フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ] – このオプションは、[MySQL レプリケーションを可能にする] が選択されておらず、[特定時点リカバリを可能にする] オプションが選択されていない場合に、デフォルトで選択されます。このオプションを使用することをお勧めしますが、ユーザはバイナリ・ログに関する制御の度合いを決定することが可能です。
- 3 各条件に対して利用可能なアクションを選択します（詳しくは、[エラー条件のデフォルト・アクションの設定（オプション）](#)を参照してください）。

各条件を使用することにより、ジョブに対して実行するアクションを選択することができます。これらとは異なるデフォルト・アクションを選択していたとしても、現在のジョブについてこの手順で指定した値に自動的に設定されます。

- [警告ありで完了 - セーブセットは保持されています] – ジョブが [警告で完了] というステータスを返し、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - [警告なしで完了 - 保存セットは保持されています] – ジョブが [バックアップが完了しました] というステータスを返し、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - [失敗 - Saveset が保持されています] – バックアップ・ジョブは、[バックアップ・ジョブ失敗] というステータスを返しますが、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
 - [失敗 - 保存セットは保持されませんでした] – バックアップ・ジョブは、[バックアップ失敗] というステータスを返し、バックアップされたオブジェクトのセーブセットは保持されません。
- 4 [Mysqldump オプション] テキスト・ボックス内に、mysqldump ユーティリティを使用するジョブをリストします。

このオプションはダッシュ (-) またはダブルダッシュ (--) で開始する必要があります。また、(; | < >) などの文字を含めることはできません。

これらのオプションはまず mysqldump コマンドに追加され、次にプラグインが内部的に生成したオプションが続きます。そのため、ここに入力したオプションが内部的に生成されたオプションと矛盾している場合、プラグインが生成したオプションが優先されます。

ジョブが失敗するようなエラーを mysqldump オプションが検出した場合、このエラーはジョブ・ログ内でエラー・ログ・メッセージに埋め込まれます。

以前、このタスクを実行するために MySQL オプション・ファイルをセットアップした場合、このテキスト・ボックスに入力したオプションは、オプション・ファイルに指定したオプションへ追加されます。本プラグインに既存の MySQL オプション・ファイルを無視させたい場合は、このテキスト・ボックスの先頭に --no-defaults と入力します。

使用しているバージョンでサポートされている mysqldump オプションについて詳しくは、利用可能な MySQL ドキュメンテーションを参照してください。

△ 警告： この機能に、--routines (-R) または --triggers オプションを使用しないよう注意してください（このオプションを使用すると、データベース・テーブルのバックアップは完了するが、リストアが失敗するなど、バックアップの正常完了を阻害する場合があります）。データベースのバックアップに必要なストアド・プロシージャやトリガがある場合、本プラグインは --routines や --triggers オプションとともに mysqldump コマンドを内部生成します。

- 5 [保存] をクリックして、セットを保存します。
- 6 [新規セットの作成] ダイアログで、セットの名前を指定して、[保存] をクリックします。

名前には英数字と英数字以外の文字を使用できますが、英語以外の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

既存のバックアップ・オプション・セットを使用するには

[プラグイン・オプション] リストで、使用する既存のバックアップ・オプション・セットを選択します。

MySQL Enterprise バックアップ用バックアップ・オプションの設定

バックアップする目的のアイテムを選択した状態で、実行するバックアップ・タイプを選択したり、失敗した場合に異なる動作を選択することも出来ます。

バックアップ・オプション・セットを作成するには

- 1 [プラグイン・オプション] リストの隣りにある **[新規作成]** をクリックします。
- 2 利用可能なオプションを選択します。
 - **[フル・バックアップ]** (デフォルト設定) – 現在の MySQL インスタンス内で設定されたすべてのデータベースおよびテーブルをバックアップするには、このオプションを選択します。
 - **[増分バックアップ]** – 最後のフルまたは増分バックアップ以降に変更が加わったデータ (InnoDB テーブル用) またはテーブル全体 (非 InnoDB テーブル用) のみをバックアップする場合、このオプションを選択します。
- 3 **[保存]** をクリックして、セットを保存します。
- 4 **[新規セットの作成]** ダイアログで、セットの名前を指定して、**[保存]** をクリックします。

名前には英数字と英数字以外の文字を使用できますが、英語以外の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

既存のバックアップ・オプション・セットを使用するには

[プラグイン・オプション] リストで、使用する既存のバックアップ・オプション・セットを選択します。

ジョブのファイナライズと実行

バックアップ・ジョブをファイナライズおよび実行するには、以下の手順に従います。

- 1 [スケジュール]、[ターゲット・ストレージ]、および [詳細設定] リストを使って、その他の必要なオプションを設定します。
- 2 **[保存]** または **[保存 & 実行]** の、どちらか適切な方をクリックします。

① **ヒント:** すでに作成、保存しているジョブを実行するには、[ナビゲーション] パネルで **[ジョブ定義管理]** を選択し、目的のジョブを選択して、**[今すぐ実行]** をクリックします。

データのリストア

- データのリストア – 概要
- MySQL におけるデータのリストア
- 高度な MySQL Standard/Community 用リストア手順

データのリストア – 概要

このセクションでは、プラグインのリストア・プロセスと利用可能なすべての機能について説明します。さらに、MySQL Standard/Community 用リストア・シナリオ例および MySQL Enterprise バックアップ用リストア・シナリオ例では、さまざまなリストア・タイプの例が用意されています。これらのセクションを熟読し、さまざまなリストア・タイプに対して利用可能な機能と適用方法について確認するようお勧めします。

MySQL Standard/Community に利用可能なリストア方法の確認

リストアを正常に実行するには、利用可能なリストア・タイプについてよく理解する必要があります。

フル・リストアまたは個々のデータベース / テーブル・コピーのみのリストア

Plug-in for MySQL でフル・バックアップまたは個々のデータベース / テーブル・コピーのみのバックアップを実行すると、MySQL の `mysqldump` ユーティリティによって、テーブルの作成とデータ追加に使用される SQL ステートメントがバックアップ・メディアに直接送信されます。プラグインがこれらのいずれかの形式で作成されたバックアップをリストアするときに、SQL ステートメントがバックアップ・メディアから直接読み取られ、自動的に実行されます。

増分または差分リストア

プラグインは増分または差分バックアップを実行するときに、MySQL のバイナリ・ログ・インデックスを使用して、バックアップ・メディアにコピーする必要があるバイナリ・ログを判断します。これらのバックアップをリストアすると、バイナリ・ログがテンポラリ・ディレクトリ (`NETVAULT_HOME/tmp/MySQL`) にリストアされます。その後、`mysqlbinlog` によって、バイナリ・ログに記録された各トランザクション用の SQL ステートメントが生成され、自動的に実行されます。このプロセスを「バイナリ・ログの適用」と言います。

増分および差分リストアの実行中に、バイナリ・ログに記録されたすべてのトランザクションを適用することも、特定時点までのトランザクションを適用 (PIT リカバリ) することもできます。PIT リ

カバリは、開発者が誤ってテーブルを削除したり誤った更新を実行するなど、データ損傷の直前の時点でリカバリする場合に便利です。

時間に基づく特定時点 (PIT) リカバリ

PIT リカバリは、増分または差分リストアの実行中にリストア対象のバイナリ・ログを使用して実行できます。時間に基づく PIT リカバリは、データの損傷時刻が分かっている場合に便利です。たとえば、開発者が午前 06:00:00 にテーブルを削除した場合、中止時刻を午前 05:55:00 に設定して PIT リカバリを実行できます。

通常、時間に基づく PIT リカバリは、1 つの手順から成るプロセスです。

[バイナリ・ログをリストアし適用する] ([オプション] タブの [バイナリ・ログをリストアし適用する] を選択) の対象として選択したバイナリ・ログを増分または差分バックアップからリストアし、不要なトランザクションの直前の中止時刻を指定します。

位置に基づく特定時点 (PIT) リカバリ

データが損傷した実際の時刻が分からない場合、またはより正確なリカバリを必要とする場合、位置に基づく特定時点リカバリを使用します。たとえば、開発者がデータベースからテーブルを削除したが、テーブルを削除した正確な時刻が分からない場合は、位置に基づく PIT リカバリを使用します。

位置に基づく PIT リカバリは、3 つの手順から成るプロセスです。

- 1 [オプション] タブの [テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する] オプションを選択して、増分または差分バックアップから MySQL サーバ上のテンポラリ・ディレクトリにバイナリ・ログをリストアします。
- 2 MySQL の `mysqlbinlog` ユーティリティを使用して、不要なトランザクションの位置を特定します。詳しくは、『MySQL リファレンス・マニュアル』の「任意時点のリカバリ」セクションを参照してください。
- 3 同じ増分または差分バックアップを再度リストアします。ただし、今度は [テンポラリ・ディレクトリからのバイナリ・ログを適用する] リストア・オプションを選択し、不要なトランザクションの直前の停止位置を指定します。

MySQL におけるデータのリストア

Plug-in for MySQL を使って標準リストアを実行するには、以下のセクションで説明する手順に従います。

- リストアするデータの選択
- リストア・オプションの設定
- ジョブのファイナライズと実行
- MySQL Standard/Community 用リストア・シナリオ例
- MySQL Enterprise バックアップ用リストア・シナリオ例

リストアするデータの選択

リストアするデータを選択するには

- 1 [ナビゲーション] パネルで、[リストア・ジョブ作成] をクリックします。
- 2 [リストア・ジョブ作成 - セーブセットの選択] ページで、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択します。

- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、[クライアント]、[日付]、[ジョブ ID] リストを使用します。

表にはセーブセット名（ジョブ・タイトルとセーブセット ID）、作成日時、およびサイズが表示されます。デフォルトでこのリストは、セーブセット名のアルファベット順にソートされています。

- 4 セーブセットの表で、適切な項目を選択します。

セーブセットを選択すると、セーブセットのジョブ ID、ジョブ・タイトル、NetVault Backup サーバ名、データをバックアップしたクライアント名、セーブセットの作成に使用されたプラグイン、セーブセットの作成日時、セーブセットのリタイア設定、増分バックアップかどうか、アーカイブかどうか、およびセーブセットのサイズなどの情報が [セーブセット情報] に表示されます。

- 5 続行するには、[次へ] をクリックします。

- 6 [セレクションセット作成] ページで、リストアするデータを選択します。

リストアの対象として選択可能な最初のノードは、リカバーするバックアップのタイプに基づいて異なります：

- **フル・バックアップまたは個々のデータベース/テーブル・コピーのみのバックアップ** ルート・ノードが [全てのデータベース] として表示されます。これは、実際のデータベース/テーブル・データがバックアップに含められたからです。

① **重要：**ルート・ノードは [全てのデータベース] という名前ですが、ターゲットの MySQL インスタンスに対して現存するデータベースをすべて包括していません。このノードを選択すると、バックアップ・ジョブとして実際に選択されたデータ・アイテムのみがリストアされます（つまり、リストア対象としてこのノードを選択すると、MySQL インスタンス内に現存するすべてのデータベースがリストアされず、単にバックアップに含めたアイテムのみがリストアされます）。

- **増分または差分バックアップ** ルート・ノードは [バイナリ・ログ] として表示されます。これは、この形式のバックアップに、前のバックアップの実行以降に行われたトランザクション（バイナリ・ログ）が含まれているからです。

- 7 アイテム別のリストアを行うには、ルート・ノードをダブルクリックして開き、バックアップに含められた個々のデータベースを表示します。

また、個々のデータベースを開いて、テーブルを表示し選択することもできます。

① **重要：**MySQL は、データベース情報の格納にさまざまなファイル形式を使用します。リストアしたデータベースが確実に機能するよう、必ずリストア・プロセスに .frm ファイルを含めるよう注意してください。

リストア・オプションの設定

[MySQL Standard/Community] または [MySQL Enterprise バックアップ] オプションのいずれを使用するかによって、[オプション] タブに各オプションが表示されます。

- [MySQL Standard/Community 用リストア・オプションの設定](#)
- [MySQL Enterprise バックアップ用リストア・オプションの設定](#)

MySQL Standard/Community 用リストア・オプションの設定

[セレクション・セット作成] ページで、[プラグイン・オプションの編集] をクリックして、[特定時点リカバリを可能にする] タブおよび [リストア先] タブで以下のパラメータを設定します。リストアに選択されたバックアップ・タイプによって、さまざまなオプションが表示されます。

- [フルまたは個々のデータベース・リストア・オプション](#)
- [増分または差分データベース・リストア・オプション](#)

フルまたは個々のデータベース・リストア・オプション

フル・バックアップまたは個別データベース / テーブルのコピーのみバックアップのいずれかをリストアするには

- 1 以下のガイドラインを利用して、[リカバリ時] タブで利用可能なオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]** – MySQL サーバ上の MySQL バイナリ・ログ・ディレクトリに保持されているバイナリ・ログを使用して、選択したデータ・オブジェクトについて**特定時点形式**のリストアを実行する場合に選択します。このオプションを選択すると、このタブ上のすべてのオプションが使用可能になります。
 - **[Point In Time (特定時点) タイプ]** – このセクションで特定時点リカバリに利用可能な形式を選択します。
 - **[時間に基づく PIT] (デフォルト選択)** – このオプションを選択して、選択したデータを指定した時間へリストアします (**時間に基づく特定時点 (PIT) リカバリ**で詳説)。このオプションを選択すると、**[時間に基づく PIT の詳細]** セクションが有効になります。
 - **[位置に基づく PIT の詳細]** – このオプションを選択して、**選択したデータを不要なトランザクションの直前に存在した特定の停止位置**へリストアします (**位置に基づく特定時点 (PIT) リカバリ**で詳説)。このオプションを選択すると、**[位置に基づく PIT の詳細]** セクションが有効になります。
 - **[時間に基づく PIT の詳細] フレーム** – **[時間に基づく PIT]** を選択した場合、以下の利用可能なオプションを選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** – 不要なトランザクションの前に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より後に実行されたトランザクションはすべて失われます。付随する **[中止日 / 時間]** フィールドに、目的の日付と時刻 (24 時間形式) を指定します。
 - **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]** – 不要なトランザクションの後に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より前に実行されたトランザクションはすべて失われます。付随する **[開始日 / 時間]** フィールドに、目的の日付と時刻 (24 時間形式) を指定します。特定の開始日と時刻に加え、トランザクションの中止日と時刻を設定することもできます。
 - **[なし] (デフォルト選択)** – **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]** に指定した日付と時刻の後に実行されたすべてのトランザクションをリストアする場合は、このラジオ・ボタンを選択したままにします。
 - **[具体的な日付]** – 特定の時間範囲に実行されたトランザクションのみを含めるには、このラジオ・ボタンを選択します。付随する時刻および日付フィールドに目的の中止時刻を入力します (24 時間形式)。

① **重要:** リストアされたバイナリ・ログと現在のバイナリ・ログの両方に対して PIT リカバリを有効にした場合、中止時刻が、リストアされたバイナリ・ログまたは現在のバイナリ・ログのどちらにあるかを特定する必要はありません。MySQL は、指定された時刻で自動的に中止 / 開始し、指定された最終的な中止時点より後のバイナリ・ログをすべて無視します。

これらのオプションを両方有効にすることもできます。特定の時間範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、2011 年 1 月 29 日の午前 11:00 から午前 11:15 までの間に収集されたデータが不要の場合、**[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** オプションを有効にし、**[中止日 / 時間]** として 11:00 – 2011 年 1 月 29 日を入力します。また、**[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]** オプションを選択し、**[開始日 / 時間]** として 11:15 – 2011 年 1 月 29 日を入力します。この結果、2011 年 1 月 29 日の 11:00 から 11:15 までの間に実行されたすべてのトランザクションがリストアから除外されます。

- **【位置に基づく PIT の詳細】** フレーム — **【位置に基づく PIT】** を選択した場合、以下の利用可能なオプションを選択します。
 - **【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** — 不要なトランザクションの**前**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**後**に実行されたトランザクションはすべて失われます。このオプションには、以下の関連オプションがあります。
 - **【停止位置】** — このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**前**の位置を入力します。たとえば、不要なトランザクションの位置が 805 の場合、804 を入力します。
 - **【終了位置を含むバイナリ・ログ】** — このドロップダウンを使用して、**【停止位置】** フィールドに指定した停止位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合（またはここに目的のファイルが表示されない場合）、**【その他】** オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** — 不要なトランザクションの**後**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**前**に実行されたトランザクションはすべて失われます。このオプションにも、以下の関連オプションがあります。
 - **【開始位置】** — このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**後**の位置を入力します。たとえば、不要なトランザクションの位置が 805 の場合、806 を入力します。
 - **【開始位置を含むバイナリ・ログ】** — このドロップダウンを使用して、**【開始位置】** フィールドに指定した開始位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合（またはここに目的のファイルが表示されない場合）、**【その他】** オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **【停止位置】: [なし]** (デフォルト選択) — **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** に指定した**開始位置**の**後**に実行された**すべての**トランザクションをリカバリするには、このラジオ・ボタンを選択したままにします。
 - **【停止位置】: [具体的な位置]** — バイナリ・ログの特定の位置範囲に実行されたトランザクションのみを含めるには、このラジオ・ボタンを選択します。**【具体的な位置】** オプションに付随するフィールドに目的の停止位置を入力し、**【終了位置を含むバイナリ・ログ】** ドロップダウン・リストで適切なバイナリ・ログ・ファイルを選択します（別のファイルを使用する場合は、このドロップダウンから**【その他】**を選択し、付随するテキスト・ボックスにファイル名を指定します。**【開始位置】**に指定した位置から**【具体的な位置】** フィールドに指定した位置までの間に実行されたトランザクションのみがリストアの対象となります。
- ① **重要:** これらのオプションを両方有効にすることもできます。特定の位置範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、位置 805 から位置 810 までの間に収集されたデータに不要なトランザクションが含まれている場合、**【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** オプションを有効にし、**【停止位置】** として 805 を入力します（また付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します）。また、**【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** オプションを有効にし、**【開始位置】** として 810 を入力します（また付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します）。この結果、指定したバイナリ・ログ・ファイルの 805 から 810 までの間に記録されたすべてのトランザクションがリストアから除外されます。停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定する必要があります。

2 以下のガイドラインに従い、[リストア先] タブで利用可能なオプションを選択します。

- **同一 MySQL インスタンスへのリストアを実行する場合** — リストアのターゲットがバックアップされた元のインスタンスと同じ場合には、これらのフィールドは空白のままにします。NetVault Backup はログインする際、[設定] ダイアログ内のこれらの値セットを使用します（詳しくは、[プラグインの設定](#)を参照してください）。
- **別の MySQL インスタンスへのリストアを実行する場合** — 選択したデータのリストアを別のインスタンスへ移動する場合には、新しいインスタンスへのリストア・アクセスを許可するために [ユーザ名] と [パスワード] の各フィールドにログイン情報を入力する必要があります。最後に、新しいインスタンスに設定された NetVault Backup 名を [インスタンス名] フィールドに入力します（[設定] ウィンドウで [MySQL インスタンス名] に設定した名前を入力します。詳しくは、[プラグインの設定](#)を参照してください）。

① **重要:** リストアを別の MySQL インスタンスへ移動する前に、異なる MySQL サーバへのリストアを参照して詳細な手順を確認する必要があります。

増分または差分データベース・リストア・オプション

増分または差分バックアップをリストアするには

1 以下のガイドラインを利用して、[リカバリ時] タブで利用可能なオプションを選択します。

- **[PIT リカバリを実行する]** — 選択したデータ・アイテムについて**特定時点形式**のリストアを実行するには、このオプションを選択します。このオプションを選択すると、このタブ上のすべてのオプションが使用可能になります。

増分および差分リストアでは、リストアを完了するためにバイナリ・ログが使用されます。このため、この形式のバックアップをリストアする場合は、選択したデータベースに関連するバイナリ・ログをどのようにリカバリするかを決定する必要があります。以下の方法の1つを選択します。

- **[バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）]** — 1つのリストア・ジョブで、バックアップ・デバイスからのバイナリ・ログをリストアし、かつバイナリ・ログに記録されたトランザクションを適用する場合に選択します。現在 MySQL バイナリ・ログ・ディレクトリに保存されているバイナリ・ログを使用して**特定時点リカバリ**を実行する場合、**[現在のバイナリ・ログを含む]** チェック・ボックスを選択します。この処理は、増分 / 差分バックアップに保存されたバイナリ・ログのトランザクションをリストアして適用した後に実行されます。
- **[テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する]** — このバイナリ・ログのリストア方法を選択した場合、リストア・ジョブによって、選択した増分 / 差分バックアップに関連するバイナリ・ログが MySQL サーバ上の**テンポラリー・ディレクトリ** (NETVAULT_HOME/tmp/MySQL/) にリストアされます。これにより、**mysqlbinlog** ユーティリティを使用して、リカバリされたログを調べ、データが損傷した時刻 / 位置を特定できるようになります。
- **[テンポラリー・ディレクトリからのバイナリ・ログを適用する]** — 前回、**[テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する]** オプションを使用し、**mysqlbinlog** ユーティリティを使用してリストアから除外する損傷データの時刻 / 位置を特定する場合、このオプションを選択します。これにより、テンポラリー・ディレクトリにリストアされたバイナリ・ログが適用されます。現在 MySQL バイナリ・ログ・ディレクトリに保存されているバイナリ・ログを使用して**特定時点リカバリ**を実行する場合、**[現在のバイナリ・ログを含む]** チェック・ボックスを選択します。この処理は、テンポラリー・ディレクトリに保存されているバイナリ・ログのトランザクションをリストアして適用した後に実行されます。
- **[Point In Time (特定時点) タイプ] オプション** — **[PIT リカバリを実行する]** オプションを有効にした場合、利用可能な PIT リカバリ形式を選択する必要があります。
 - **[時間に基づく PIT] (デフォルト選択)** — このオプションを選択して、選択したデータを指定した時間へリストアします (**時間に基づく特定時点 (PIT) リカバリ**で詳説)。このオプションを選択すると、**[時間に基づく PIT の詳細]** セクションが有効になります。

- **【位置に基づく PIT の詳細】** — このオプションを選択して、**選択したデータを不要なトランザクションの直前に存在した特定の停止位置へリストア**します（**位置に基づく 特定時点 (PIT) リカバリ**で詳述）。このオプションを選択すると、**【位置に基づく PIT の詳細】** セクションが有効になります。
- **【時間に基づく PIT の詳細】 フレーム** — **【時間に基づく PIT】** を選択した場合、以下の利用可能なオプションを選択します。
 - **【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** — 不要なトランザクションの**前に**起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より**後に**実行されたトランザクションはすべて失われます。付随する **【中止日／時間】** フィールドに、目的の日付と時刻（24 時間形式）を指定します。
 - **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** — 不要なトランザクションの**後に**起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より**前に**実行されたトランザクションはすべて失われます。付随する **【開始日／時間】** フィールドに、目的の日付と時刻（24 時間形式）を指定します。特定の開始日と時刻に加え、トランザクションの中止日と時刻を設定することもできます。
 - **【なし】（デフォルト選択）** — **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** に指定した日付と時刻の後に実行されたすべてのトランザクションをリストアする場合は、このラジオ・ボタンを選択したままにします。
 - **【具体的な日付】** — 特定の時間範囲に実行されたトランザクションのみを含めるには、このラジオ・ボタンを選択します。付随する時刻および日付フィールドに目的の中止時刻を入力します（24 時間形式）。
- ① **重要：**これらのオプションを両方有効にすることもできます。特定の時間範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、2011 年 1 月 29 日の午前 11:00 から午前 11:15 までの間に収集されたデータが不要の場合、**【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** オプションを有効にし、**【中止日／時間】** として 11:00 — 2011 年 1 月 29 日を入力します。また、**【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** オプションを選択し、**【開始日／時間】** として 11:15 — 2011 年 1 月 29 日を入力します。この結果、2011 年 1 月 29 日の 11:00 から 11:15 までの間に実行されたすべてのトランザクションがリストアから除外されます。
- **【位置に基づく PIT の詳細】 フレーム** — **【位置に基づく PIT】** を選択した場合、以下の利用可能なオプションを選択します。
 - **【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** — 不要なトランザクションの**前に**起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**後に**実行されたトランザクションはすべて失われます。このオプションには、以下の関連オプションがあります。
 - **【停止位置】** — このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**前の**位置を入力します。たとえば、不要なトランザクションの位置が 805 の場合、804 を入力します。
 - **【終了位置を含むバイナリ・ログ】** — このドロップダウンを使用して、**【停止位置】** フィールドに指定した停止位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合（またはここに目的のファイルが表示されない場合）、**【その他】** オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** — 不要なトランザクションの**後に**起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**前に**実行されたトランザクションはすべて失われます。このオプションにも、以下の関連オプションがあります。
 - **【停止位置】** — このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**前の**位置を入力します。たとえば、不要なトランザクションの位置が 805 の場合、804 を入力します。
 - **【終了位置を含むバイナリ・ログ】** — このドロップダウンを使用して、**【停止位置】** フィールドに指定した停止位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合（またはここに目的のファイルが表示されない場合）、**【その他】** オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。

- **【開始位置】** — このフィールドに、バイナリ・ログ内の不要なトランザクションよりも後の位置を入力します。たとえば、不要なトランザクションの位置が 805 の場合、806 を入力します。
- **【開始位置を含むバイナリ・ログ】** — このドロップダウンを使用して、**【開始位置】** フィールドに指定した開始位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合（またはここに目的のファイルが表示されない場合）、**【その他】** オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
- **【停止位置】**: **【なし】**（デフォルト選択） — **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** に指定した**開始位置**の後に実行されたすべてのトランザクションをリカバリするには、このラジオ・ボタンを選択したままにします。
- **【停止位置】**: **【具体的な位置】** — バイナリ・ログの特定の位置範囲に実行されたトランザクションのみを含めるには、このラジオ・ボタンを選択します。**【具体的な位置】** オプションに付随するフィールドに目的の停止位置を入力し、**【終了位置を含むバイナリ・ログ】** ドロップダウン・リストで適切なバイナリ・ログ・ファイルを選択します（別のファイルを使用する場合は、このドロップダウンから**【その他】**を選択し、付随するテキスト・ボックスにファイル名を指定します。**【開始位置】** に指定した位置から**【具体的な位置】** フィールドに指定した位置までの間に実行されたトランザクションのみがリストアの対象となります。

① **重要**：これらのオプションを両方有効にすることもできます。特定の位置範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、位置 805 から位置 810 までの間に収集されたデータに不要なトランザクションが含まれている場合、**【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** オプションを有効にし、**【停止位置】** として 805 を入力します（また付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します）。また、**【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** オプションを有効にし、**【開始位置】** として 810 を入力します（また付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します）。この結果、指定したバイナリ・ログ・ファイルの 805 から 810 までの間に記録されたすべてのトランザクションがリストアから除外されます。停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定する必要があります。

2 以下のガイドラインに従い、**【リストア先】** サブタブで利用可能なオプションを選択します。

このタブには、3 つのフィールドで構成される **【リストア先詳細】** セクションが表示されます。これらのフィールドにアカウント情報を入力して、MySQL のターゲット・インスタンスへのリストア・アクセスを許可します。目的のリストア・タイプに応じて、以下のようにオプションを使用します。

- **同一 MySQL インスタンスへのリストアを実行する場合** — リストアのターゲットがバックアップされた元のインスタンスと同じ場合には、これらのフィールドはブランクのままにします。NetVault Backup はログインする際、**【設定】** ダイアログ内のこれらの値セットを使用します（詳しくは、[プラグインの設定](#)を参照してください）。
- **別の MySQL インスタンスへのリストアを実行する場合** — 選択したデータのリストアを別のインスタンスへ移動する場合には、新しいインスタンスへのリストア・アクセスを許可するために **【ユーザ名】** と **【パスワード】** の各フィールドにログイン情報を入力する必要があります。最後に、新しいインスタンスに設定された NetVault Backup 名を **【インスタンス名】** フィールドに入力します（**【設定】** ウィンドウで **【MySQL インスタンス名】** に設定した名前を入力します。詳しくは、[プラグインの設定](#)を参照してください）。

① **重要**：リストアを別の MySQL インスタンスへ移動する前に、**異なる MySQL サーバへのリストア**を参照して詳細な手順を確認する必要があります。

MySQL Enterprise バックアップ用リストア・オプションの設定

[セクション・セット作成] ページで、[プラグイン・オプションの編集] をクリックして、[オプション] タブで適切なパラメータを設定します。

① **重要:** リストアを実行する前に、[MySQL Enterprise バックアップ] オプションを使用して作成されたフル・バックアップ内に含まれるすべてのデータを（少なくとも一時的に）格納するのに十分な空き容量がデフォルトの NetVault Backup の Temp ディレクトリにあることを確認します。[一般] オプションを使用して、十分な空き容量を持つ格納場所にデフォルト設定を変更することができ、これによりマップ済みドライブやネットワーク・ファイル・システム（NFS）、または SMB マウント等も使用できるようになります。[ナビゲーション] パネルで [設定変更] をクリックして、[サーバ設定] をクリックして、次に [システムおよびセキュリティ] セクションの [一般] をクリックします。

- **フル・リストア** – 利用可能なオプションを選択します。
 - [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成]（デフォルト選択）– このオプションを選択して、MySQL サーバのデータ・リポジトリ・ディレクトリ階層に対応する一時位置へフル・バックアップをリストアします。このオプションを使用する場合は、どのバックアップをリストアするか識別していることが前提です。まだの場合は、以下の 2 つのオプションを使用することができます。
 - [フル・バックアップ・イメージを Temp ファイルにリストア] – バックアップのコンテンツをリストして、次のオプションを実行する必要があるか識別する場合はこのオプションを選択します。
 - [Temp ファイルから Raw フル・バックアップを抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] – 前述の [フル・バックアップ・イメージを Temp ファイルにリストア] オプションを使用した結果について、どのバックアップをリストアする必要があるか識別するには、このオプションを選択します。このオプションにより、フル・バックアップが MySQL サーバのデータ・リポジトリ・ディレクトリ階層に対応する一時格納場所へリストアされます。
 - [MySQL サーバをシャットダウンし、準備済みフル・バックアップを MySQL サーバ・リポジトリへコピー・バック] – MySQL サーバをシャットダウンし、リストア済みコンテンツを一時格納場所から元の位置へコピーバックする準備が整った際、このオプションを選択します。
 - [バックアップ・イメージの検証] – 抽出したデータに対して検証コマンドの実行を本プラグインに指示する場合、このチェックボックスを選択します。
 - [バックアップ・イメージのリスト] – 出力ログ内のバックアップ・コンテンツをリストするには、このオプションを選択します。
- **増分リストア** – 利用可能なオプションを選択します。
 - [増分バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成]（デフォルト選択）– 増分バックアップをリストアするには、このオプションを選択します。このオプションを使用する場合は、どのバックアップをリストアするか識別していることが前提です。まだの場合は、以下の 2 つのオプションを使用することができます。
 - [増分バックアップ・イメージを Temp ファイルにリストア] – バックアップのコンテンツをリストして、次のオプションを実行する必要があるか識別する場合はこのオプションを選択します。
 - [Temp ファイルから増分バックアップを抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] – 前述の [増分バックアップ・イメージを Temp ファイルにリストア] オプションを使用した結果について、どのバックアップをリストアする必要があるか識別するには、このオプションを選択します。
 - [MySQL サーバをシャットダウンし、準備済みフル・バックアップを MySQL サーバ・リポジトリへコピー・バック] – MySQL サーバをシャットダウンし、リストア済みコンテンツを一時格納場所から元の位置へコピーバックする準備が整った際、このオプションを選択します。
 - [バックアップ・イメージの検証] – 抽出したデータに対して検証コマンドの実行を本プラグインに指示する場合、このチェックボックスを選択します。
 - [バックアップ・イメージのリスト] – 出力ログ内のバックアップ・コンテンツをリストするには、このオプションを選択します。

ジョブのファイナライズと実行

最終ステップには、[スケジュール]、[ソース・オプション]、および [詳細設定] ページの追加オプション設定、ジョブの実行、および [ジョブ・ステータス] と [ログ参照] ページからの進捗状況の監視が含まれています。これらのページとオプションは、すべての NetVault Backup プラグインに共通しています。詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

リストア・ジョブをファイナライズおよび実行するには、以下の手順に従います。

- 1 [Ok] をクリックして設定を保存し、[次へ] をクリックします。
- 2 デフォルト設定を使用しない場合は、[ジョブ名] に、ジョブの名前を指定します。
進捗状況を監視する際にジョブを識別しやすくするため、具体的な名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、英語以外の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

① 重要: ジョブ・タイトルには、ターゲット OS のファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windows では /、\、*、@ などの文字は使用できません (Plug-in for MySQL がデータを一時的にリストアするために、[ジョブ・タイトル] と同じ名前のフォルダを作成しようとするからです)。
- 3 [クライアント指定] リストで、データをリストアするマシンを選択します。

① ヒント: [選択] をクリックして、[クライアント指定選択] ダイアログから適切なクライアントを検索、選択することもできます。
- 4 [スケジュール]、[ソース・オプション]、および [詳細設定] リストを使って、その他の必要なオプションを設定します。
- 5 [実行] をクリックして、スケジュール設定のためにジョブを送信します。
[ジョブステータス] ページでは進捗をモニタしたり、[ログ参照] ページではログを参照表示することができます。詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

MySQL Standard/Community 用リストア・シナリオ例

障害またはデータ損傷から正しくリカバリするには、ジョブの設定時に、リストア対象として選択するデータおよび [オプション] タブのオプションに関してさまざまな設定を行う必要があります。以降のセクションでは、さまざまなタイプのリカバリ例を示し、必要となるオプションについて説明します。

- フル・バックアップのみによるリストア・シナリオ
- フルおよび増分バックアップによるリストア・シナリオ
- フルおよび差分バックアップによるリストア・シナリオ

フル・バックアップのみによるリストア・シナリオ

以下の例で、MySQL 管理者は毎晩午後 11 : 00 にフル・バックアップを実行するバックアップ・ポリシーを設定しました。

フル・バックアップのリストアと時間に基づく特定時点リカバリ

データベース管理者は月曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の月曜日午前 6:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

方法 1：誤った SQL ステートメントの前をリカバリする

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。つまり、データベース管理者は日曜日の夜に実行されたフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 [日曜日の夜からのフル・リストアを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。
 - [現在のバイナリ・ログで PIT リカバリを実行する] — このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [時間に基づく PIT] — リストア・タイプとして選択しました。
 - [誤った／不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[中止日／時間] を 5:59、2011 年 1 月 31 日（月曜日の日付の午前 6:00 の 1 分前）に設定しました。
- 3 ジョブを開始します。

方法 2：誤った SQL ステートメントの前および後をリカバリする

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、誤った／不良の SQL ステートメントの後の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。

- 1 [日曜日の夜からのフル・リストアを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。
 - [現在のバイナリ・ログで PIT リカバリを実行する] — このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [時間に基づく PIT] — リストア・タイプとして選択しました。
 - [誤った／不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[中止日／時間] を 5:59、2011 年 1 月 31 日（月曜日の日付の午前 6:00 の 1 分前）に設定しました。
 - [誤った／不良の SQL ステートメントの後に、リカバリを可能にする] — Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を [開始日／時間] オプションに入力しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、[開始日／時間] オプションで [なし] ラジオ・ボタンを選択しました。

フル・リストアと位置に基づく特定時点リカバリ

データベース管理者は月曜日の午前 9:00 に、ユーザが Orders テーブルで table not found（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の月曜日午前 6:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

方法 1：誤った SQL ステートメントの前をリカバリする

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は日曜日のフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 **現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する** – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが `MYSQLSVR-bin.000009` バイナリ・ログの 805 の位置にあることを特定しました。
- 2 **[日曜日の夜からのフル・リストアを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 3 **リストア関連の [オプション] タブのオプション設定** – データベース管理者は以下のオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]** – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - **[位置に基づく PIT]** – リストア・タイプとして選択しました。
 - **[誤った/不良の SQL ステートメントの前に、リカバリを可能にする]** – このオプションを選択し、**[停止位置]** を 804 (mysqlbinlog で特定した位置の前の位置) に設定しました。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前 (`MYSQLSVR-bin.000009` など) をテキスト・ボックスに入力しました。

① **重要：** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

- 4 **ジョブを開始します。**

方法 2：誤った SQL ステートメントの前および後をリカバリする

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができず。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は日曜日のフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 **現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する** – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが `MYSQLSVR-PM-bin.000009` バイナリ・ログの 805 の位置にあることを特定しました。
- 2 **[日曜日の夜からのフル・リストアを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 3 **リストア関連の [オプション] タブのオプション設定** – データベース管理者は以下のオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]** – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。

- **[位置に基づく PIT]** – リストア・タイプとして選択しました。
- **[誤った／不良の SQL ステートメントの前に、リカバリを可能にする]** – このオプションを選択し、**[停止位置]** を 804 (mysqlbinlog で特定した位置の前の位置) に設定しました。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前 (MYSQLSVR-PM-bin.000009 など) をテキスト・ボックスに入力しました。
- **[誤った／不良の SQL ステートメントの後に、リカバリを可能にする]** – このオプションを選択し、**[開始位置]** を 806 (mysqlbinlog で特定した位置の後の位置) に設定しました。**[開始位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前 (MYSQLSVR-bin.000009 など) をテキスト・ボックスに入力しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、**[停止位置]** オプションで **[なし]** ラジオ・ボタンを選択しました。

① **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

4 ジョブを開始します。

フルおよび増分バックアップによるリストア・シナリオ

DBA はフル・バックアップを毎週日曜日の午後 11 時に、増分バックアップを月～土曜日の午後 11 時に実行するバックアップ・ポリシーを作成しました。DBA は増分バックアップを実行しているため、各増分バックアップの実行後バイナリ・ログは削除されます。これにより、全体的なバックアップ時間は短くなりますが、リストアにはより多くの時間と手順が必要になります。

フルおよび増分リストアのみ

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の木曜日午前に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、最後の増分バックアップ、つまり水曜日の夜に実行されたバックアップの時点までを完全にリカバリすることを決定しました。

手順 1 : 日曜日からのフル・リストア

- 1 **[日曜日の夜からのフル・バックアップを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** – どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待つ**

手順 2 : 月曜日からの増分リストア

- 1 **[月曜日の夜からの増分バックアップを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** – どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待つ。**

手順 3 : 火曜日からの増分リストア

- 1 **[火曜日の夜からの増分バックアップを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。

- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：水曜日からの増分リストア

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始します。

フル・リストアと時間に基づく特定時点リカバリ

以降の例では、フル / 増分バックアップ・シナリオを取り上げます。データベース管理者はデータを特定時点にリカバリしようと考えています。

方法 1：リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が水曜日の夜 8:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するように指定しました。
 - [時間に基づく PIT] – リストア・タイプとして選択しました。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 19:59、2011 年 1 月 31 日 (水曜日の日付の午後 8:00 の 1 分前) に設定しました。
- 3 ジョブを開始します。

方法 2：リストアされたバイナリ・ログのみを使用して誤った/不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、午後 8 時に drop table コマンドが実行された直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。

- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 4：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するように指定しました。
 - [時間に基づく PIT] – リストア・タイプとして選択しました。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 19:59、2011 年 1 月 31 日 (水曜日の日付の午後 8:00 の 1 分前) に設定しました。
 - [誤った/不良の SQL ステートメントの後に、リカバリを可能にする] – Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を [開始日/時間] オプションに入力しました。最後に、バックアップに含まれるバイナリ・ログ・ファイルを最後までリカバリするために、[開始日/時間] オプションで [なし] ラジオ・ボタンを選択しました。
- 3 ジョブを開始します。

方法 3：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が木曜日の午前 06:00:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 4：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するように指定しました。
 - [現在のバイナリ・ログを含む] – 水曜日にバックアップが完了してから drop table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にしました。
 - [時間に基づく PIT] – リストア・タイプとして選択しました。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 5:59、2011 年 1 月 31 日 (木曜日の日付の午前 6:00 の 1 分前) に設定しました。
- 3 ジョブを開始します。

方法 4：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定しました。
 - [現在のバイナリ・ログを含む] – 水曜日にバックアップが完了してから drop table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にしました。
 - [時間に基づく PIT] – リストア・タイプとして選択しました。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 5:59、2011 年 1 月 31 日 (木曜日の日付の午前 6:00 の 1 分前) に設定しました。
 - [誤った/不良の SQL ステートメントの後に、リカバリを可能にする] – Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を [開始日/時間] オプションに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、[開始日/時間] オプションで [なし] ラジオ・ボタンを選択しました。
- 3 ジョブを開始します。

フル・リストアと位置に基づく特定時点リカバリ

以降の例では、フル / 増分バックアップ・シナリオを取り上げます。データベース管理者は、より確実な方法で時刻を特定し、データを特定時点にリカバリしようと考えています。これは、MySQL バイナリ・ログ・ファイル内で特定した位置の値を使用して行います。

方法 1：リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9：00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜

8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：バックアップされたバイナリ・ログをリストアして誤った SQL ステートメントの位置を特定する

この手順では、水曜日の夜の差分バックアップに記録されたバイナリ・ログについてテンポラリー・ロケーションへのリストアのみを実行します。これにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する] – 水曜日の夜の増分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択しました。
 - [時間に基づく PIT] – データベース管理者は、[Point In Time (特定時点) タイプ] としてこのオプションが選択されていることを確認しましたが、[時間に基づく PIT の詳細] セクションに表示されるその他すべてのオプションは選択解除したままにしました。
- 3 ジョブを開始して完了するまで待つ。

手順 5：リストアされたバイナリ・ログで dropped table コマンドの位置を特定する

リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する。リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバのテンポラリ・ロケーションにリストアされた MYSQLSVR-bin.000009 バイナリ・ログの 805 の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順 6：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された増分バックアップを選択する。データベース管理者は再度 [リストア・ジョブ作成 - セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定。データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]。このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [テンポラリ・ディレクトリからのバイナリ・ログを適用する]。一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択しました。リストアしたバイナリ・ログ・ファイルで dropped table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする]。このオプションを選択し、[停止位置] を 804（バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置）に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（MYSQLSVR-bin.000009）を選択しました。

- 3 ジョブを開始します。

方法 2：リストアされたバイナリ・ログのみを使用して誤った/不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9：00 に、ユーザが Orders テーブルで table not found（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する]。まず、データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。

- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：バックアップされたバイナリ・ログをリストアして誤った SQL ステートメントの位置を特定する

この手順では、水曜日の夜の差分バックアップに記録されたバイナリ・ログについてテンポラリー・ロケーションへのリストアのみを実行します。これにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 [水曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する] - 水曜日の夜の増分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択しました。
 - [時間に基づく PIT] – データベース管理者は、[Point In Time (特定時点) タイプ] としてこのオプションが選択されていることを確認しましたが、[時間に基づく PIT の詳細] セクションに表示されるその他すべてのオプションは**選択解除**したままにしました。
- 3 ジョブを開始して完了するまで待つ。

手順 5：リストアされたバイナリ・ログで dropped table コマンドの位置を特定する

リストアされたバイナリ・ログに対して `mysqlbinlog` ユーティリティを使用する – リストアしない `drop table` SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は、`drop table` ステートメントが、MySQL サーバのテンポラリー・ロケーションにリストアされた `MYSQLSVR-bin.000009` バイナリ・ログの 805 の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順 6 : 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された増分バックアップを選択する — データベース管理者は再度 [リストア・ジョブ作成 — セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
 - 2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] — このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [テンポラリー・ディレクトリからのバイナリ・ログを適用する] — 一連の操作の最後の手順で、テンポラリー・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択しました。リストアしたバイナリ・ログ・ファイルで dropped table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[停止位置] を 804 (バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置) に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリー・ディレクトリにリストアしたバイナリ・ログ・ファイル (MYSQLSVR-bin.000009) を選択しました。
 - [誤った/不良の SQL ステートメントの後に、リカバリを可能にする] — データベース管理者はこのオプションを選択し、[開始位置] を 806 (現在のバイナリ・ログ・ファイルの、mysqlbinlog で特定した dropped table コマンドの位置の後にある位置) に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリー・ディレクトリにリストアしたバイナリ・ログ・ファイル (MYSQLSVR-bin.000009) を選択しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、[開始日/時間] オプションで [なし] ラジオ・ボタンを選択しました。
- ① **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

3 ジョブを開始します。

方法 3 : リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が木曜日の午前 06:00:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下に手順を示します。

手順 1 : 日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。

- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：現在のバイナリ・ログで dropped table コマンドの位置を特定する

現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが MYSQLSVR-bin.000009 バイナリ・ログの 805 の位置にあることを特定しました。

手順 5：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された増分バックアップを選択する – データベース管理者は再度 [リストア・ジョブ作成 – セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。
 - [現在のバイナリ・ログを含む] – データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の増分バックアップの後に実行されたすべてのデータベース・トランザクションを適用するよう指定しました。これにより、水曜日の夜に増分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[停止位置] を 804 (バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置) に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを [OTHER FILE] に設定し、現在のバイナリ・ファイルの名前 (MYSQLSVR-bin.000009 など) をテキスト・ボックスに入力しました。

方法 4：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が木曜日の午前 06:00:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。これを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：月曜日からの増分リストア

- 1 [月曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 3：火曜日からの増分リストア

- 1 [火曜日の夜からの増分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 4：現在のバイナリ・ログで dropped table コマンドの位置を特定する

現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが MYSQLSVR-bin.000009 バイナリ・ログの 805 の位置にあることを特定しました。

手順 5：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された増分バックアップを選択する – データベース管理者は再度 [リストア・ジョブ作成 – セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。

2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。

- **[PIT リカバリを実行する]** – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
- **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** – バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。
- **[現在のバイナリ・ログを含む]** – データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の増分バックアップの後に実行されたすべてのデータベース・トランザクションを適用するよう指定しました。これにより、水曜日の夜に増分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- **[誤った/不良の SQL ステートメントの前に、リカバリを可能にする]** – このオプションを選択し、**[停止位置]** を 804 (バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の**前**にある位置) に設定しました。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、現在のバイナリ・ファイルの名前 (MYSQLSVR-bin.000009 など) をテキスト・ボックスに入力しました。
- **[誤った/不良の SQL ステートメントの後に、リカバリを可能にする]** – データベース管理者はこのオプションを選択し、**[開始位置]** を 806 (現在のバイナリ・ログ・ファイルの、mysqlbinlog で特定した dropped table コマンドの位置の**後**にある位置) に設定しました。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、現在のバイナリ・ファイルの名前 (MYSQLSVR-bin.000009 など) をテキスト・ボックスに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、**[停止位置]** オプションで **[なし]** ラジオ・ボタンを選択しました。

① **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

フルおよび差分バックアップによるリストア・シナリオ

DBA はフル・バックアップを毎週日曜日の午後 11 時に、差分バックアップを月～土曜日の午後 11 時に実行するバックアップ・ポリシーを作成しました。DBA は差分バックアップを実行するため、このバックアップの各フォーム後にバイナリ・ログが保持されます。そのため、バックアップは長くなりますが、総合的なリストアは高速になります。

フルおよび差分リストアのみ

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の木曜日午前に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、最後の差分バックアップ、つまり水曜日の夜に実行されたバックアップの時点までを完全にリカバリすることを決定しました。

手順 1 : 日曜日からのフル・リストア

- 1 **[日曜日の夜からのフル・バックアップを選択する]** – まず、データベース管理者は **[リストア・ジョブ作成 – セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** – どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待つ。**

手順 2：水曜日からの増分リストア

- 1 [水曜日の夜からの差分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – [オプション] タブのどのオプションも使用しません。
- 3 ジョブを開始します。

① **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

フル・リストアと時間に基づく特定時点リカバリ

以降の例では、フル / 差分バックアップ・シナリオを取り上げます。データベース管理者はデータを特定時点にリカバリしようと考えています。

方法 1：リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9：00 に、ユーザが Orders テーブルで table not found（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が水曜日の夜 8:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。このことから、以下の手順を実行します。

手順 1：日曜日の夜からのフル・リストアを実行

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの差分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

① **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。

- [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定しました。
- [時間に基づく PIT] – リストア・タイプとして選択しました。
- [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 19:59、2011 年 1 月 31 日 (水曜日の日付の午後 8:00 の 1 分前) に設定しました。

3 ジョブを開始します。

方法 2：リストアされたバイナリ・ログのみを使用して誤った/不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、午後 8 時に drop table コマンドが実行された直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの差分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

① 重要：差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません (水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます)。

- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定しました。
 - [時間に基づく PIT] – リストア・タイプとして選択しました。
 - [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[中止日/時間] を 19:59、2011 年 1 月 31 日 (水曜日の日付の午後 8:00 の 1 分前) に設定しました。

- **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** – Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を **【開始日／時間】** オプションに入力しました。最後に、リストアしたバイナリ・ログ・ファイルを最後までリカバリするために、**【開始日／時間】** オプションで **【なし】** ラジオボタンを選択しました。

3 ジョブを開始します。

方法 3：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った／不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9：00 に、ユーザが Orders テーブルで **table not found**（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 06:00:00** に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が木曜日の午前 06:00:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 **【日曜日の夜からのフル・バックアップを選択する】** – まず、データベース管理者は **【リストア・ジョブ作成 – セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** – どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待つ。**

手順 2：水曜日からの時間に基づく PIT リストア

- 1 **【水曜日の夜からの差分バックアップを選択する】** – まず、データベース管理者は **【リストア・ジョブ作成 – セーブセットの選択】** ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

① **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は**月曜日および火曜日の夜の差分バックアップをリストアする必要がありません**（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

- 2 **リストア関連の【オプション】タブのオプション設定** – データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】** – このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - **【バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）】** – このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定しました。
 - **【現在のバイナリ・ログを含む】** – 水曜日にバックアップが完了してから drop table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にしました。
 - **【時間に基づく PIT】** – リストア・タイプとして選択しました。
 - **【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** – このオプションを選択し、**【中止日／時間】** を 5:59、2011 年 1 月 31 日（木曜日の日付の午前 6:00 の 1 分前）に設定しました。

3 ジョブを開始します。

方法 4：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする — どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：水曜日からの時間に基づく PIT リストア

- 1 [水曜日の夜からの差分バックアップを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

❶ **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません (水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます)。

- 2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] — このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にしました。
 - [バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)] — このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定しました。
 - [現在のバイナリ・ログを含む] — 水曜日にバックアップが完了してから drop table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にしました。
 - [時間に基づく PIT] — リストア・タイプとして選択しました。
 - [誤った / 不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[中止日 / 時間] を 5:59、2011 年 1 月 31 日 (木曜日の日付の午前 6:00 の 1 分前) に設定しました。
 - [誤った / 不良の SQL ステートメントの後に、リカバリを可能にする] — Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を [開始日 / 時間] オプションに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、[開始日 / 時間] オプションで [なし] ラジオ・ボタンを選択しました。
- 3 ジョブを開始します。

フル・リストアと位置に基づく特定時点リカバリ

以降の例では、フル / 増分バックアップ・シナリオを取り上げます。データベース管理者は、より確実な方法で時刻を特定し、データを特定時点にリカバリしようと考えています。これは、MySQL バイナリ・ログ・ファイル内で特定した位置の値を使用して行います。

方法 1：リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする – どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ

手順 2：バックアップされたバイナリ・ログをリストアして誤った SQL ステートメントの位置を特定する

この手順では、水曜日の夜の差分バックアップに記録されたバイナリ・ログについてテンポラリー・ロケーションへのリストアのみを実行します。これにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 [水曜日の夜からの差分バックアップを選択する] – まず、データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する] – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - [テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する] – 水曜日の夜の差分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択しました。
 - [時間に基づく PIT] – データベース管理者は、[Point In Time (特定時点) タイプ] としてこのオプションが選択されていることを確認しましたが、[時間に基づく PIT の詳細] セクションに表示されるその他すべてのオプションは選択解除したままにしました。
- 3 ジョブを開始して完了するまで待つ

手順 3：リストアされたバイナリ・ログで dropped table コマンドの位置を特定する

リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は、drop table ステートメント

が、MySQL サーバのテンポラリ・ロケーションにリストアされた `MYSQLSVR-bin.000009` バイナリ・ログの `805` の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順 4：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された差分バックアップを選択する — データベース管理者は再度 [リストア・ジョブ作成 — セーブセットの選択] タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

① **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

- 2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。

- [PIT リカバリを実行する] — このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
- [テンポラリ・ディレクトリからのバイナリ・ログを適用する] — 一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択しました。リストアしたバイナリ・ログ・ファイルで `dropped table` コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
- [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[停止位置] を `804`（バイナリ・ログの、`mysqlbinlog` で特定した `dropped table` コマンドの位置の前にある位置）に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（`MYSQLSVR-bin.000009`）を選択しました。

- 3 ジョブを開始します。

方法 2：リストアされたバイナリ・ログのみを使用して誤った/不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザが `Orders` テーブルで `table not found`（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、水曜日の夜 8:00 に開発者が無意識のうちに `Orders` テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、`drop table` コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、`Orders` テーブルが削除された時点の後の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする — どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：バックアップされたバイナリ・ログをリストアして誤った SQL ステートメントの位置を特定する

この手順では、水曜日の夜の差分バックアップに記録されたバイナリ・ログについてテンポラリー・ロケーションへのリストアのみを実行します。これにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 **【水曜日の夜からの差分バックアップを選択する】** – まず、データベース管理者は **【リストア・ジョブ作成 – セーブセットの選択】** ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の【オプション】タブのオプション設定** – データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】** – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - **【テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する】** – 水曜日の夜の差分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択しました。
 - **【時間に基づく PIT】** – データベース管理者は、**【Point In Time（特定時点）タイプ】** としてこのオプションが選択されていることを確認しましたが、**【時間に基づく PIT の詳細】** セクションに表示されるその他すべてのオプションは**選択解除**したままにしました。
- 3 **ジョブを開始して完了するまで待つ**

手順 3：リストアされたバイナリ・ログで dropped table コマンドの位置を特定する

リストアされたバイナリ・ログに対して **mysqlbinlog ユーティリティ** を使用する – リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバのテンポラリー・ロケーションにリストアされた **MYSQSVR-bin.000009** バイナリ・ログの 805 の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順 4：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 **水曜日の夜に実行された差分バックアップを選択する** – データベース管理者は再度 **【リストア・ジョブ作成 – セーブセットの選択】** タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
 - ① **重要：** 差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は**月曜日および火曜日の夜の差分バックアップをリストアする必要がありません**（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。
- 2 **リストア関連の【オプション】タブのオプション設定** – データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】** – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
 - **【テンポラリー・ディレクトリからのバイナリ・ログを適用する】** – 一連の操作の最後の手順で、テンポラリー・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択しました。リストアしたバイナリ・ログ・ファイルで dropped table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使われるようにこのオプションを選択します。

- **【誤った／不良の SQL ステートメントの前に、リカバリを可能にする】** — このオプションを選択し、**【停止位置】** を 804 (バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置) に設定しました。**【終了位置を含むバイナリ・ログ】** ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル (MYSQLSVR-bin.000009) を選択しました。
- **【誤った／不良の SQL ステートメントの後に、リカバリを可能にする】** — データベース管理者はこのオプションを選択し、**【開始位置】** を 806 (現在のバイナリ・ログ・ファイルの、mysqlbinlog で特定した dropped table コマンドの位置の後にある位置) に設定しました。**【終了位置を含むバイナリ・ログ】** ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル (MYSQLSVR-bin.000009) を選択しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、**【停止位置】** オプションで **【なし】** ラジオ・ボタンを選択しました。

① **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

3 ジョブを開始します。

方法 3 : リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った／不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザが Orders テーブルで table not found (テーブルが見つかりません) エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が木曜日の午前 06:00:00 にテーブルを削除した直前の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下に手順を示します。

手順 1 : 日曜日からのフル・リストア

- 1 **【日曜日の夜からのフル・バックアップを選択する】** — まず、データベース管理者は **【リストア・ジョブ作成 — セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** — どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待つ。**

手順 2 : 現在のバイナリ・ログで dropped table コマンドの位置を特定する

現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する — リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが MYSQLSVR-bin.000009 バイナリ・ログの 805 の位置にあることを特定しました。

手順 3 : 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 **水曜日の夜に実行された差分バックアップを選択する** — データベース管理者は再度 **【リストア・ジョブ作成 — セーブセットの選択】** タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

- ① **重要**：差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は**月曜日**および**火曜日**の夜の差分バックアップをリストアする必要が**ありません**（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

2 リストア関連の [オプション] タブのオプション設定 — データベース管理者は以下のオプションを選択します。

- [PIT リカバリを実行する] — このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
- [バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）] — バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。
- [現在のバイナリ・ログを含む] — データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の差分バックアップの後に実行されたすべてのデータベース・トランザクションを適用するよう指定しました。これにより、水曜日の夜に差分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- [誤った/不良の SQL ステートメントの前に、リカバリを可能にする] — このオプションを選択し、[停止位置] を 804（バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置）に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを [OTHER FILE] に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。

方法 4：リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9：00 に、ユーザが Orders テーブルで table not found（テーブルが見つかりません）エラーに遭遇しているという通知を受けました。調査の結果、木曜日の午前 06:00:00 に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される直前の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の後の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下に手順を示します。

手順 1：日曜日からのフル・リストア

- 1 [日曜日の夜からのフル・バックアップを選択する] — まず、データベース管理者は [リストア・ジョブ作成 — セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする — どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待つ。

手順 2：現在のバイナリ・ログで dropped table コマンドの位置を特定する

現在のバイナリ・ログに対して mysqlbinlog ユーティリティを使用する — リストアしない drop table SQL ステートメントの位置を特定するために、この作業を NetVault Backup の外で実行します。この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください。この処理で、データベース管理者は drop table ステートメントが MYSQLSVR-bin.000009 バイナリ・ログの 805 の位置にあることを特定しました。

手順 3：位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜に実行された差分バックアップを選択する – データベース管理者は再度 [リストア・ジョブ作成 – セーブセットの選択] タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

① **重要：**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されるため、データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません（水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます）。

- 2 リストア関連の [オプション] タブのオプション設定 – データベース管理者は以下のオプションを選択します。

- [PIT リカバリを実行する] – このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にしました。
- [バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）] – バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。
- [現在のバイナリ・ログを含む] – データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の差分バックアップの後に実行されたすべてのデータベース・トランザクションを適用するよう指定しました。これにより、水曜日の夜に差分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- [誤った／不良の SQL ステートメントの前に、リカバリを可能にする] – このオプションを選択し、[停止位置] を 804（バイナリ・ログの、mysqlbinlog で特定した dropped table コマンドの位置の前にある位置）に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを [OTHER FILE] に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。
- [誤った／不良の SQL ステートメントの後に、リカバリを可能にする] – データベース管理者はこのオプションを選択し、[開始位置] を 806（現在のバイナリ・ログ・ファイルの、mysqlbinlog で特定した dropped table コマンドの位置の後にある位置）に設定しました。[終了位置を含むバイナリ・ログ] ドロップダウンを [OTHER FILE] に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、[停止位置] オプションで [なし] ラジオ・ボタンを選択しました。

① **重要：**停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

MySQL Enterprise バックアップ用リストア・シナリオ例

障害またはデータ損傷から正しくリカバリするには、ジョブの設定時に、リストア対象として選択するデータおよび [オプション] タブのオプションに関してさまざまな設定を行う必要があります。

- フル・バックアップのみによるリストア・シナリオ
- フルおよび増分バックアップによるリストア・シナリオ

フル・バックアップのみによるリストア・シナリオ

- 1 リストアする準備済みフル・バックアップを生成するには、[オプション] タブで選択した [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] オプションを選択したジョブを実行します。
- 2 MySQL をシャットダウンし、MySQL サーバ・リポジトリに準備済みバックアップをコピーするには、[オプション] タブで選択した [MySQL サーバをシャットダウンし、準備済みフル・バックアップを MySQL サーバ・リポジトリへコピー・バック] オプションを選択したジョブを実行します。
- 3 コマンド・プロンプトで利用可能なコマンドを入力して MySQL を再起動します。

フルおよび増分バックアップによるリストア・シナリオ

- 1 リストアする準備済みフル・バックアップを生成するには、[オプション] タブで選択した [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] オプションを選択したジョブを実行します。
- 2 必要な増分バックアップを準備済みフル・バックアップへ、それらがバックアップされた順に適用するには、[オプション] タブで選択した [増分バックアップをリストア、抽出し、Temp ディレクトリ内の準備済みフル・バックアップに適用] オプションを選択したジョブを実行します。
- 3 MySQL をシャットダウンし、MySQL サーバ・リポジトリに準備済みバックアップをコピーするには、[オプション] タブで選択した [MySQL サーバをシャットダウンし、準備済みフル・バックアップを MySQL サーバ・リポジトリへコピー・バック] オプションを選択したジョブを実行します。
- 4 コマンド・プロンプトで利用可能なコマンドを入力して MySQL を再起動します。

高度な MySQL Standard/Community 用リストア手順

このセクションでは、[MySQL Standard/Community] オプション用プラグインを使用して実行することができるその他のリストア操作について説明します。

- リストア中にデータベース名を変更する
- 同じサーバ上の別の MySQL インスタンスへリストアする
- 異なる MySQL サーバへのリストア

リストア中にデータベース名を変更する

NetVault Backup では、バックアップした MySQL データベースを選択し、リストア時に名前を変更して、既存のデータベースが上書きされないよう設定することができます。これは、既存のデータベースのコピーを作成する場合に便利です。これを実行するには、以下のセクションで説明する手順に従います。

- ① **重要:** リストアではデータベース全体の名称変更のみを行います。個別のテーブルの名称を変更しようとするとエラー・メッセージが表示されます。

リストア時の名前変更を実行する前に、この操作に関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスは、フル・バックアップまたは個々のデータベース / テーブル・コピーのみのバックアップに制限されます。
- 増分および差分リストアの実行中に行うことはできません。
- 別の MySQL インスタンスまたは MySQL サーバへのリストアと組み合わせて使用できます。

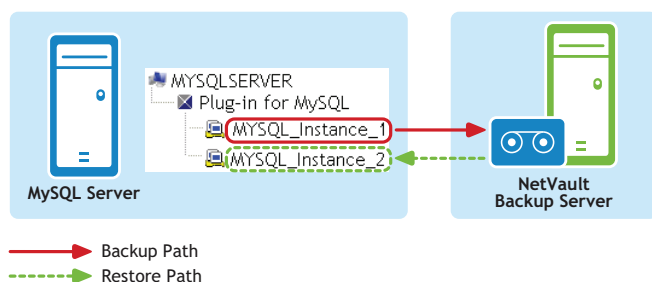
リストア時にデータベース名を変更するには

- 1 [ナビゲーション] パネルで [リストア・ジョブ作成] をクリックして、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択し、適切なセーブセットを選択して [次へ] をクリックします。
詳しくは、[リストアするデータの選択](#)を参照してください。
- 2 [セレクションセット作成] ページで、名前を変更するデータベースを選択します。
- 3 [アクション] リストから、[名前変更] を選択します。
- 4 [名前変更 / 移動] ダイアログの [名前変更] ボックスに新しい名前を入力して、[OK] をクリックします。
データベース・アイテムでは、追加情報が括弧内に表示されています。
- 5 「MySQL におけるデータのリストア」の説明に従い、リストアを続行します。

同じサーバ上の別の MySQL インスタンスへリストアする

Plug-in for MySQL によるバックアップを同じ MySQL サーバ・マシンであるが、そこで構成された MySQL の別のインスタンスへリストアします。

図 1. この形式のリストアでは、1つの MySQL インスタンスからバックアップされたデータを別のインスタンスへリカバリします。



これを実行するには、以下のセクションで説明する手順に従います。

既知の制限事項 / 想定される用途

リストア時のパス名移動を実行する前に、この操作に関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスには、フル、増分、差分バックアップ、および個々のデータベース / テーブル・コピーのみのバックアップを含めることができます。
- リストア先インスタンスに適用できるのは、増分または差分バックアップからリストアされたバイナリ・ログのみです（ソース・インスタンスの現在のバイナリ・ログをリストア先インスタンスに適用することはできません）。

前提条件

このタイプのリストアを設定して実行するには、以下の前提条件を満たす必要があります。

- **既存のマシンとターゲット・マシンのインストール構成を同一にする** – MySQL に関して、両方のマシンが以下のように設定されている必要があります。
 - 同じオペレーティング・システムがインストールされていること
 - 同じバージョンの MySQL がインストールされていること

- インストール / ベース・ディレクトリが同一であること
- MySQL データ・ディレクトリが同一であること
- Plug-in for MySQL で新しいターゲット・インスタンスで構成 — 新しい MySQL インスタンスを追加するには、[プラグインの設定](#)で説明した手順を正しく実行する必要があります（ターゲット・インスタンスを [NetVault Backup 選択] ページの [Plug-in for MySQL] ノードの下で表示してアクセス可能にします）。

リストアの設定と開始

前提条件が満たされた状態で、以下の手順に従ってこの形式の移動リストア・ジョブを設定します。

リストアを実行するには

- 1 [ナビゲーション] パネルで、[リストア・ジョブ作成] をクリックします。
- 2 [リストア・ジョブ作成 — セーブセットの選択] ページで、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択します。
- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、[クライアント]、[日付]、[ジョブ ID] リストを使用します。
表にはセーブセット名（ジョブ・タイトルとセーブセット ID）、作成日時、およびサイズが表示されます。デフォルトでこのリストは、セーブセット名のアルファベット順にソートされています。
- 4 セーブセットの表で、適切な項目を選択します。
セーブセットを選択すると、セーブセットのジョブ ID、ジョブ・タイトル、NetVault Backup サーバ名、データをバックアップしたクライアント名、セーブセットの作成に使用されたプラグイン、セーブセットの作成日時、セーブセットのリタイア設定、増分バックアップかどうか、アーカイブかどうか、およびセーブセットのサイズなどの情報が [セーブセット情報] に表示されます。
- 5 続行するには、[次へ] をクリックします。
- 6 [セレクションセット作成] ページで、リストアするデータを選択します。
バックアップ対象だった個別の MySQL インスタンスを表示して、選択ツリーから目的のデータ・アイテムを探し、それらを選択します。
- 7 利用可能なデータベースを選択したら [プラグイン・オプションの編集] をクリックし、[リストア先] タブを選択します。
- 8 [リストア先詳細] セクションで、以下の情報を入力します。
 - [ユーザ名] — ターゲット MySQL インスタンスにアクセスするためのログオン・アカウントを入力します。
 - [パスワード] — ログオン・アカウントに関するパスワードを入力します。
 - [インスタンス名] — NetVault Backup の構成時の設定に基づいて、新しい MySQL ターゲット・インスタンスの NetVault Backup 名前を入力します（[設定] ウィンドウの [MySQL インスタンス名] フィールドでインスタンスに設定した値（詳しくは、[プラグインの設定](#)を参照してください））。
- 9 必要に応じて、[リカバリ時] タブで使用可能なその他のオプションを設定できますが、この形式のリストアを実行するために必ずしも必要ではありません。詳しくは、「[リストア・オプションの設定](#)」を参照してください。
- 10 [Ok] をクリックして設定を保存し、[次へ] をクリックします。
- 11 デフォルト設定を使用しない場合は、[ジョブ名] に、ジョブの名前を指定します。
進捗状況を監視する際にジョブを識別しやすくするため、具体的な名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、英語以外の文字を含めることはできません。Linux の場

合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

- ① **重要:** ジョブ・タイトルには、ターゲット OS のファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windows では /、\、*、@ などの文字は使用できません (Plug-in for MySQL がデータを一時的にリストアするために、[ジョブ・タイトル] と同じ名前のフォルダを作成しようとするからです)。

12 **【クライアント指定】** リストで、データをリストアするマシンを選択します。

- ① **ヒント:** **【選択】** をクリックして、**【クライアント指定選択】** ダイアログから適切なクライアントを検索、選択することもできます。

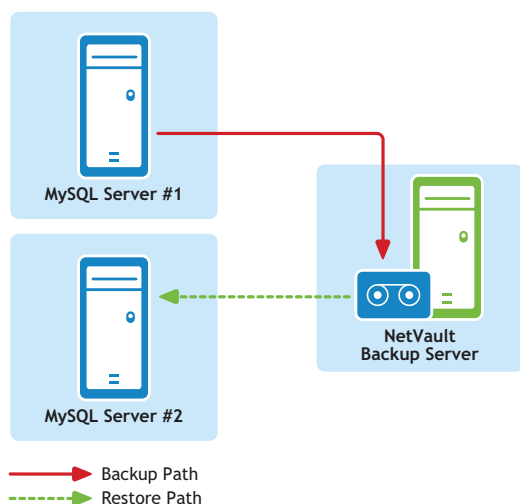
13 **【スケジュール】**、**【ソース・オプション】**、および **【詳細設定】** リストを使って、その他の必要なオプションを設定します。

14 **【実行】** をクリックして、スケジュール設定のためにジョブを送信します。

異なる MySQL サーバへのリストア

Plug-in for MySQL でデータベースまたは個々のテーブルを同じ MySQL サーバ上の別の MySQL インスタンスにリストアできるのと同様に、リストア・プロセス中に別の MySQL サーバをターゲットにすることができます。このようなリストアは、災害復旧オペレーションで使用されます。

図 2. この形式の移動リストアのデータ・パス例



これを実行するには、以下のセクションで説明する手順に従います。

既知の制限事項 / 想定される用途

リストアを別の MySQL サーバに移動する前に、このオペレーションに関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスには、フル、増分、差分バックアップ、および個々のデータベース / テーブル・コピーのみのバックアップを含めることができます。
- リストア先インスタンスに適用できるのは、増分または差分バックアップからリストアされたバイナリ・ログのみです (ソース・インスタンスの現在のバイナリ・ログをリストア先の MySQL インスタンスに適用することはできません)。

ソフトウェアのインストール / 設定の前提条件

このタイプのリストアを設定して実行するには、以下の前提条件を満たす必要があります。

- **既存のマシンとターゲット・マシンのインストール構成を同一にする** – MySQL に関して、両方のマシンが以下のように設定されている必要があります。
 - 同じオペレーティング・システムがインストールされていること
 - 同じバージョンの MySQL がインストールされていること
 - インストール / ベース・ディレクトリが同一であること
 - MySQL データ・ディレクトリが同一であること
- **NetVault Backup ソフトウェアおよび Plug-in for MySQL をすべてのクライアントにインストールする** – NetVault Backup (クライアントまたはサーバ・バージョン) およびプラグインを、このプロセスで使用する両方のクライアント・マシン (既存の MySQL マシンと新しいリストア・ターゲット) にインストールし、設定する必要があります。
- **すべてのクライアント・マシンを NetVault Backup サーバに追加する** – すべてのソフトウェア・インストール条件を満たした上で、NetVault Backup WebUI を使用して、ターゲットの NetVault Backup クライアント・マシン (既存の MySQL マシンおよび新しいリストア・ターゲット) を NetVault Backup サーバに追加する必要があります。
- **新しいリストア・ターゲットに MySQL インスタンスが必ず存在する** – 新しいリストア・ターゲットに MySQL インスタンスが存在している必要があります。このインスタンスが移動リストアのターゲットとなります。このインスタンスは MySQL で正しく設定および構成されるとともに、新しいリストア・ターゲットでプラグインのインストールに追加されている必要があります ([プラグインの設定](#)で説明した手順に従います)。

❶ **重要:** 新しいリストア・ターゲットのインスタンスの [設定] ウィンドウから、以下の値をメモしておきます。ユーザー名、パスワードおよびインスタンス名

移動リストアの設定中、プラグインでターゲット MySQL インスタンスへの適切なアクセスを得られるよう、これらの値を [オプション] タブ内のフィールドに入力する必要があります。

リストアの実行

すべての前提条件を満たしたら、以下の手順に従って、MySQL のバックアップを別のマシンにリストアします。

リストアを実行するには

- 1 [ナビゲーション] パネルで、[リストア・ジョブ作成] をクリックします。
- 2 [リストア・ジョブ作成 – セーブセットの選択] ページで、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択します。
- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、[クライアント]、[日付]、[ジョブ ID] リストを使用します。

表にはセーブセット名 (ジョブ・タイトルとセーブセット ID)、作成日時、およびサイズが表示されます。デフォルトでこのリストは、セーブセット名のアルファベット順にソートされています。

- 4 セーブセットの表で、適切な項目を選択します。

セーブセットを選択すると、セーブセットのジョブ ID、ジョブ・タイトル、NetVault Backup サーバ名、データをバックアップしたクライアント名、セーブセットの作成に使用されたプラグイン、セーブセットの作成日時、セーブセットのリタイア設定、増分バックアップかどうか、アーカイブかどうか、およびセーブセットのサイズなどの情報が [セーブセット情報] に表示されます。

- 5 続行するには、[次へ] をクリックします。

- 6 **[セクションセット作成]** ページで、リストアするデータを選択します。

バックアップ対象だった個別の MySQL インスタンスを表示して、選択ツリーから目的のデータ・アイテムを探し、それらを選択します。
- 7 利用可能なデータベースを選択したら **[プラグイン・オプションの編集]** をクリックし、**[リストア先]** タブを選択します。
- 8 **[リストア先詳細]** セクションで、以下の情報を入力します。
 - **[ユーザ名]** – 新しいリストア・ターゲットのターゲット・インスタンスに設定したユーザー名を入力します（**[設定]** ウィンドウの **[ユーザ名]** フィールドで設定した名前）。
 - **[パスワード]** – 新しいリストア・ターゲットのターゲット・インスタンスに設定したパスワードを入力します（**[設定]** ウィンドウの **[パスワード]** フィールドで設定したパスワード）。
 - **[インスタンス名]** – 新しいリストア・ターゲットの MySQL インスタンスの NetVault Backup 名を入力します（**[設定]** ウィンドウの **[MySQL インスタンス名]** フィールドでインスタンスに設定した値）。
- 9 必要に応じて、**[リカバリ時]** タブで使用可能なその他のオプションを設定できますが、この形式のリストアを実行するために必ずしも必要ではありません。詳しくは、「**リストア・オプションの設定**」を参照してください。
- 10 **[Ok]** をクリックして設定を保存し、**[次へ]** をクリックします。
- 11 デフォルト設定を使用しない場合は、**[ジョブ名]** に、ジョブの名前を指定します。

進捗状況を監視する際にジョブを識別しやすくするため、具体的な名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、英語以外の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にすることをお勧めします。

① **重要：** ジョブ・タイトルには、ターゲット OS のファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windows では /、\、*、@ などの文字は使用できません（Plug-in for MySQL がデータを一時的にリストアするために、**[ジョブ・タイトル]** と同じ名前のフォルダを作成しようとするからです）。
- 12 **[クライアント指定]** リストで、データをリストアするマシンを選択します。

① **ヒント：** **[選択]** をクリックして、**[クライアント指定選択]** ダイアログから適切なクライアントを検索、選択することもできます。
- 13 **[スケジュール]**、**[ソース・オプション]**、および **[詳細設定]** リストを使って、その他の必要なオプションを設定します。
- 14 **[実行]** をクリックして、スケジュール設定のためにジョブを送信します。

MySQL レプリケーションの使用

- MySQL レプリケーション環境でのプラグイン使用 – 概要
- レプリケーションのサポートの有効化
- レプリケーション・サーバのバックアップ
- レプリケーション・サーバのリストア

MySQL レプリケーション環境でのプラグイン使用 – 概要

レプリケーションを使用するときは、複製されたテーブルに対するすべての更新をマスタ・サーバで実行する必要があります。これを行わない場合は、ユーザがマスタ上のテーブルに対して行った更新と、スレーブ上のテーブルに対して行った更新との間に矛盾が生じないように常に注意する必要があります。

レプリケーションにより、堅牢性、速度、およびシステム管理上の利点がもたらされます。

- マスタ / スレーブ設定により、堅牢性が向上します。マスタで問題が発生した場合、バックアップとして機能しているスレーブに切り替えることができます。
- クライアント・クエリの処理負荷をマスタ・サーバとスレーブ・サーバに分散することにより、クライアントへの応答時間を短縮できます。SELECT クエリをスレーブに送信し、マスタのクエリ処理負荷を軽減することもできます。ただし、マスタとスレーブが同期から外れないようにするために、データを変更するステートメントについては、マスタに送信する必要があります。このロード・バランス戦略は、更新以外のクエリが中心となる場合に効果的ですが、実際はこれが標準的なケースです。
- レプリケーションのもう1つの利点は、マスタに負荷をかけることなく、スレーブ・サーバを使用してデータベース・バックアップを実行できることです。バックアップ中も、マスタは継続して更新を処理します。

Plug-in for MySQL は、単一マスタ・レプリケーション環境のバックアップとリカバリをサポートしています。

レプリケーションのサポートの有効化

レプリケーションのサポートは、**[設定]** ダイアログを使って有効にします。このダイアログへのアクセスについて詳しくは、「[プラグインの設定](#)」を参照してください。

レプリケーションを有効にするには

- **[MySQL レプリケーションを可能にする]** – このインスタンスに対してネイティブの MySQL レプリケーションを有効化するには、このチェック・ボックスを選択します。
 - **[スレーブ・インスタンス]** – MySQL レプリケーションを有効化したインスタンスをスレーブ・インスタンスとして設定する場合、このオプションを選択します。

- **[マスタ・インスタンス]** – MySQL レプリケーションを有効化したインスタンスをマスタ・インスタンスとして設定する場合、このオプションを選択します。
- **[特定時点リカバリを可能にする]** – 特定時点バックアップおよびリストアを有効化した場合、このチェック・ボックスを選択します。
- **[バイナリ・ログ・インデックス・パス]** – **[特定時点リカバリを可能にする]** チェックボックスを選択した場合、このフィールドを使用してバイナリ・ログ・インデックス・ファイルへのフル・パスを指定します。
- **[リレー・ログ・インデックス・パス]** – **[スレーブ・インスタンス]** を設定中の場合、このフィールドを利用してリレー・ログ・インデックス・ファイルへのフル・パスを入力し、バックアップに含めます。

レプリケーション・サーバのバックアップ

MySQL レプリケーション環境のバックアップは、以下の制限付きでサポートされます。

- **スレーブ・レプリケーション・サーバ** – サポートされるバックアップ・タイプは以下のとおりです。
 - [フル]
 - [増分]
 - [差分]
 - 個々のデータベース/テーブル・コピーのみ
- **マスタ・レプリケーション・サーバ** – サポートされるバックアップ・タイプは以下のとおりです。
 - 個々のデータベース/テーブル・コピーのみ

スレーブ・サーバで増分および差分バックアップを実行するには、MySQL で `--log-slave-updates` オプションを有効にする必要があります。このオプションは、スレーブの SQL スレッドによって実行された更新をそれ自体のバイナリ・ログに記録するようスレーブに指示するものです。このオプションを動作させるには、バイナリ・ログを有効にする “`--log-bin`” オプションを使用してスレーブを起動する必要があります。通常、このオプションはレプリケーション・サーバを連結するために使用されますが、バイナリ・ログ・バックアップに対して使用すると、マスタ・サーバ上のバイナリ・ログをスレーブへの適用前にパージするといった面倒なオペレーションを行わなくても、レプリケーション環境で PIT リカバリが可能になります。

レプリケーション設定のバックアップ

[リレー・ログ・インデックス・パス] オプションを使用することにより、リレー・ログ・インデックス・ファイルへのフル・パスを指定し、バックアップに含めることができます。デフォルトでは、ステータス・ファイル `master.info` および `relay-log.info` は、同一の場所に配置されています。**[リレー・ログ・インデックス・パス]** オプションを使用し、デフォルトのファイル名および位置を指定した場合、これらのファイルすべてがバックアップされ、プラグインにより自動的にスレーブ・レプリケーション・サーバにリストアされます。

レプリケーション・サーバのリストア

MySQL レプリケーション・スレーブ・インスタンスのフル、増分および差分バックアップを使用して、MySQL レプリケーション・マスタ・インスタンスの災害復旧を実行することができます。マスタのリストア後、データベース管理者は同じバックアップ・セットを使用して各スレーブ・インスタンスをマスタ・インスタンスと同レベルにリストアして、レプリケーションを再開したり、『MySQL Reference Guide』で説明されているその他の初期化方法でスレーブを再度初期化することができます。

マスタおよびスレーブ双方の個々のデータベース / テーブル・バックアップを使用して、個々のデータベース / テーブルをマスタにリストアすることができます。スレーブ上の個々のテーブルまたはデータベースを再同期する場合は、スレーブにリストアしてからスレーブとマスタを再同期するよう配慮するのではなく、MySQL のレプリケーション・プロセスを利用して再同期することをお勧めします。

フェイルオーバー・クラスタ環境でのプラグインの使用

- MySQL サーバ・フェイルオーバー・クラスタリング – 概要
- プラグインのインストール
- プラグインのライセンス
- プラグインの設定
- データのバックアップ
- データのリストア

MySQL サーバ・フェイルオーバー・クラスタリング – 概要

MySQL フェイルオーバー・クラスタリング（アクティブ/パッシブ）は、MySQL サーバ・インスタンス全体の高可用性を確保することを目的に設計されています。たとえば、フェイルオーバー・クラスタの1つのノードで、ハードウェア障害またはオペレーティング・システム障害が発生した場合、あるいは計画されたアップグレードを行う際に、クラスタ内の別のノードにフェイルオーバーするよう MySQL サーバ・インスタンスを設定できます。

フェイルオーバー・クラスタは、1つ以上のノード（ホスト）と1つ以上の共有ディスクで構成されます。IP アドレス、共有ストレージ、およびアプリケーション（この場合、MySQL）などのノードによってホスティングされるさまざまなリソースを組み合わせることでクラスタ・サービスと呼ばれるグループを構成します。ネットワーク上では、仮想サービスはアプリケーションを実行中の単一のコンピュータとして認識されますが、現在のノードが使用不可になった場合はノード間でのフェイルオーバーが可能です。

- ① **重要：** NetVault Backup の用語において、クラスタ・サービスを**仮想クライアント**と呼びます。基本的に、Plug-in for MySQL における**仮想クライアント**とは、MySQL サーバ・フェイルオーバー・クラスタ環境の**クラスタ・サービス**を指します。

Plug-in for MySQL は、MySQL サーバ・フェイルオーバー・クラスタリングをサポートしています。プラグインはフェイルオーバー・クラスタ・ネットワーク名によって、MySQL サーバ・クラスタ・サービスを管理している現在のノードを特定し、それをバックアップ対象とすることができます。

このセクションでは、プラグインの設定と使用が、フェイルオーバー・クラスタ環境と従来の環境でどのように異なるかを説明します。このセクションは、[MySQL Standard/Community] オプションを説明するセクションを反映させるため、以下のような構成になっています。

重要な注意事項

- 以降のセクションで説明していない場合、本プラグインを使用したクラスタ・データのバックアップおよびリストア手順は、従来の MySQL サーバ・データのバックアップおよびリストア手順と同様です。
- 以降のセクションでは、フェイルオーバー・クラスタ環境で本プラグインを使用する場合に必要な、MySQL 固有の設定についてのみ説明しています。NetVault Backup の **アプリケーション・クラスタ・サポート** を使用して MySQL サーバ以外の関連データ / ファイルのバックアップ / リストアを管理する設定手順については説明していません。このプロセスは本プラグイン固有のものではありません。この手順については詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください。
- 次のセクションに進む前に、『Dell NetVault Backup アドミニストレーターズ・ガイド』のすべてのクラスタ関連情報を確認し、ここで説明する内容が MySQL サーバ・フェイルオーバー・クラスタ機能とどのように関連しているかを理解しておくことを強くお勧めします。Dell

プラグインのインストール

本プラグインをインストールするには、以下の手順に従います。

インストールの前提条件

Plug-in for MySQL をクラスタ環境にインストールするには、以下の前提条件を満たす必要があります。

- **MySQL フェイルオーバー・クラスタリング環境を展開する** – 正しく構成された MySQL クラスタリング環境が必要です。
 - ① **重要:** この機能のサポートは Red Hat Clustering および Clustered Storage Suite を使用する Red Hat Enterprise Linux (RHEL) v5.x 上でテストされ、データベースのデータ・ファイルおよびログを含む共有ストレージと MySQL (v5.5) 2 ノード・クラスタ構成を採用します。クラスタリング機能を異なる構成で使用する場合、実際の運用環境に配置する前に、バックアップおよびリストアをテストする必要があります。
- **NetVault Backup サーバ・マシンを別に用意する** – NetVault Backup サーバとして使用するマシンを正しく設定して MySQL サーバ・クラスタの**外部**に設置し、クラスタ内のノード（ホスト）へのネットワーク接続を行う必要があります。

ソフトウェアのインストール

クラスタ環境へのプラグインのインストールは、従来のインストール方法とは異なっています。このプロセスは、NetVault Backup サーバに**仮想クライアント**を作成することで完成します。仮想クライアントとは、クラスタ内のノード / ホストのグループで、NetVault Backup サーバからは、1つのクライアントとして認識されます。仮想クライアントは、1つのクラスタ・サービス（MySQL クラスタ化サービスなど）をバックアップする場合に作成します。仮想クライアントの作成プロセスでは、プラグインが NetVault Backup サーバからクラスタ内の選択したノードに転送され、そこにインストールされます。

仮想クライアントの作成

前述のとおり、仮想クライアント作成プロセスは、本プラグイン固有のものではありません。この手順については詳しくは、『Dell NetVault Backup アドミニストレーターズ・ガイド』を参照してください。ただし、仮想クライアント作成プロセス中は、以下の点を考慮する必要があります。

- **仮想クライアントに名前を付ける** – NetVault Backup 仮想クライアント名として、MySQL クラスタ化サービスに割り当てられたネットワーク名を使用することを強くお勧めします。DellNetVault Backup では、仮想クライアントを参照するときに、クラスタ・サービスを管理しているノードが特定され、そのMySQL サーバ・インスタンスが表示されます（[NetVault Backup 選択] ページなど）。仮想クライアント名として MySQL クラスタ化サービス・ネットワーク名と同じ名前を設定しておけば、仮想クライアントに対応する MySQL サーバ・インスタンスをより簡単に識別できます。
- **関連するクラスタ・ノードのみを仮想クライアントに追加する** – 仮想クライアントの作成時に含めるホストを、バックアップ/リストア対象の MySQL クラスタ化サービスに関連するクラスタ内のノードに限定します。

仮想クライアントの作成後、プラグインはすべての指定クラスタ・ノードに転送され、ローカルにインストールされます。インストールされた Plug-in for MySQL を仮想クライアントから使用して、共有データをバックアップ/リストアできます（クラスタ内で共有設定したデータのバックアップおよびリストアのみ）。

プラグインのライセンス

Plug-in for MySQL をクラスタ環境で使用する場合のもう1つの違いとして、使用ライセンスがあります。本プラグインは、共有データのバックアップおよびリストアのみをサポートしています。このため、MySQL サーバ・フェイルオーバー・クラスタ環境に必要な仮想クライアントのクラスタ・アプリケーション・ライセンスは1つのみです。

ライセンス・キーの入手などの手続きについては、『Dell NetVault Backup インストレーション・ガイド』を参照してください。

プラグインの設定

このセクションでは、Plug-in for MySQL クラスタ化サービスで使用する仮想クライアントは作成済みであることを想定しています。NetVault Backup サーバで以下の手順に従って、実際の仮想クライアントを設定する必要があります。

プラグインを設定するには

- 1 NetVault Backup サーバの NetVault Backup WebUI の [ナビゲーション] パネルで、[バックアップ ジョブ作成] をクリックして、次に [セレクション] リストの隣りにある [新規作成] をクリックします。
- 2 選択ツリーで、新たに作成した仮想クライアントを開きます。
- 3 Plug-in for MySQL を開いて、[すべてのインスタンス] ノードを選択します。
- 4 [アクション] リストから、[設定] を選択します。
- 5 [設定] ダイアログで、利用可能な設定オプションを設定します。

使用可能なオプションは、[プラグインの設定](#) セクションで説明するオプションと同じです。

① 重要： [設定] ダイアログの [インスタンス名] フィールドに各クラスタ・インスタンスを追加する必要があります。インスタンスを追加するには、MySQL クラスタ・サービス名を VIRTUAL SERVER NAME\INSTANCE NAME の形式で指定します。

- 6 その他のバックアップ・ジョブの作成や、セカンダリ・ノード上で既存バックアップ・ジョブの変更が見込まれる場合は、以下の手順を実行します。
 - a プライマリ・ノードをセカンダリ・ノードにフェイルオーバーします。
 - b [ステップ 1](#) から [ステップ 5](#) を繰り返します。
 - c プライマリ・ノードにフェイル・バックします。
- 7 [OK] をクリックして設定を保存します。

データのバックアップ

仮想クライアントで使用するよう設定した Plug-in for MySQL によるバックアップは、比較的簡単です。[NetVault Backup 選択] ページで Plug-in for MySQL ノードを開き、バックアップ対象にする MySQL サーバ仮想サーバ（またはそれに含まれるアイテム）を選択します。

実際には、このページに表示されるインスタンス名は、インストール・プロセスで仮想クライアントとして設定した MySQL クラスタ化サービスです（「[仮想クライアントの作成](#)」を参照してください）。このノードでほかの MySQL クラスタ化サービスが稼働している場合、これらのインスタンスも Plug-in for MySQL ノード内に表示されます。このような他のインスタンスのデータをバックアップ対象として選択しないよう注意してください。

- ① データのバックアップまたはリストアを実行する際は、ノードの1つからではなく、必ず仮想クライアントを使用してプロセスを実行する必要があります。ノードを展開して階層をドリル・ダウンしていくと、MySQL クラスタ化サービスが表示され、ノードがアクティブかどうかによって、さらにアイテムをドリル・ダウンして選択することができます。ログ情報の管理目的上、システムがこのインスタンスを使用している可能性があるため、このレベルではいかなる処理も実行しないよう注意してください。

データのリストア

仮想クライアントへのデータのリストアは、従来の NetVault Backup クライアントに対してリストアを実行する場合と同様の方法で行います。Plug-in for MySQL を使用したリストアに使用可能なすべてのオプションは、フェイルオーバー・クラスタリング環境でも使用できます。また、データも同様の方法で選択します。両者の違いは、[リストア・ジョブ作成 - セーブセットの選択] ページで、リストア可能な仮想クライアントのバックアップが、バックアップ中に使用していた個々の NetVault Backup クライアントまたはノードではなく、仮想クライアント名下に表示される点だけです。リストア・ジョブを開始すると、NetVault Backup はすべてのメンバー・クライアントと通信し、フェイルオーバー・クラスタを管理しているマシンを特定し、このマシンをリストア対象として指定します。

さらに、NetVault Backup 仮想クライアントを、非クラスタ化（スタンドアローン）NetVault Backup クライアントへリストアすることもできます。

このユーザーズ・ガイドで説明しているリストア実行手順は、仮想クライアントのリカバリでも使用できます。NetVault Backup 仮想クライアントのリストア手順については、「[データのリストア](#)」セクションを参照してください。NetVault Backup 仮想クライアントを NetVault Backup スタンドアローンのクライアントにリストアするには、「[異なる MySQL サーバへのリストア](#)」セクションで説明している手順に従います。

トラブルシューティング

このセクションでは一般的なエラーとその解決方法について記述します。この表に記載されていないエラーが発生した場合は、NetVault Backup ログから MySQL エラー番号を取得し、MySQL のドキュメンテーションで関連するトラブルの解決手段を参照してください。

表 2. トラブルシューティング

エラー・メッセージ	説明
NetVault Backup 10.x サービス (netvault-pgsql) が Windows で開始しない	Windows イベント・ビューアに以下のメッセージが表示されているかどうかを確認します。PDT FATAL: lock file "postmaster.pid" already exists NetVault Backup 10.x は PostgreSQL データベースを使用しています。PostgreSQL データベースが開始しない場合、NetVault Backup を開始することはできません。この問題に対処するには、ログで参照されている場所にある「postmaster.pid」を削除して、NetVault Backup サーバを再起動します。詳しくは https://support.software.dell.com/netvault-backup/kb/122475 を参照してください。
<ul style="list-style-type: none"> バックアップ・レコードの追加に失敗しました バックアップ・インデックスをデータベースに書き込むことができませんでした <p>これらのメッセージは、選択されたデータのバックアップは完了したが、NetVault Backup によってジョブのインデックス情報がデータベースに適切に追加されなかったことを示します。このインデックス情報が追加されていないと、データは正しくリストアされません。</p>	<p>方法 1: NetVault Backup WebUI の [デバイス管理] ページに移動して、ジョブが対象とするメディアのスキャンを実行します。NetVault Backup では、バックアップ・ジョブのインデックス情報は NetVault データベースとバックアップ対象メディアの双方に保存されます。このスキャンを実行することで、インデックス情報が、NetVault データベースに書き込まれます。これを確認するには、[リストア・ジョブ作成 - セーブセットの選択] ページを開いて対象のジョブを見つけます。ジョブが参照でき、リストア・ジョブも設定できる場合には、スキャン処理によってこの問題は解決されました。</p> <p>方法 2: 方法 1 が正しく実行できない場合は、バックアップ・ジョブを再実行する必要があります。</p>
バックアップがレプリケーション・エラーを伴って失敗しました。	バックアップが「レプリケーション・スレーブ・サーバの開始に失敗しました」に類似したメッセージを伴って失敗する場合、 [MySQL レプリケーションを可能にする] チェックボックスは選択したが、レプリケーションを設定していない可能性を示します。この問題を修正するには、 [設定] ダイアログの [MySQL レプリケーションを可能にする] チェックボックスをクリアするか、レプリケーションを設定してからバックアップ・ジョブを再実行する必要があります。設定の更新について詳しくは、 プラグインの設定 を、レプリケーションについて詳しくは、 MySQL レプリケーションの使用 を参照してください。

Dell はお客様の声を大切にし、常に製品やサービスの向上に努めております。詳しくは、www.software.dell.com を参照してください。

Dell へのお問い合わせ

テクニカル・サポート :
オンライン・サポート

販売製品に関するご質問 :
03-5908-3511

電子メール :
Sales.JP@quest.com

テクニカル・サポート用リソース

テクニカル・サポートは、有効なメンテナンス契約が付いた Dell ソフトウェアをご購入のお客様、およびトライアル版をご使用のお客様がご利用いただけます。サポート・ポータルにアクセスするには、<https://support.software.dell.com/jp> に移動してください。

サポート・ポータルには、問題を素早く自力で解決するために役立つ自己支援ツールが用意されており、1 年中毎日 24 時間ご利用いただけます。また、このポータルのオンライン・サービス・リクエスト・システムを利用して、製品サポート・エンジニアに直接アクセスすることもできます。

このサイトでは、以下の作業を行えます。

- サービス・リクエスト（案件）の作成、更新、管理
- Knowledge Base 記事の参照
- 製品に関するお知らせの入手
- ソフトウェアのダウンロードトライアル版のソフトウェアについては、[Trial Downloads](#)（トライアル版のダウンロード）に移動してください。
- 入門ビデオの表示
- コミュニティでのディスカッション