

MySQL Enterprise Backup ユーザーズガイド (バージョン 3.11.1)

概要

これは MySQL Enterprise Backup 製品のユーザースガイドです。このマニュアルには、MySQL データベースをバックアップしてリストアする手順が記載されています。バックアップの時間とストレージオーバーヘッドを最小限に抑えて、バックアップ操作中も継続してデータベースを使用できるようにする手法を説明しています。選択したデータベースやテーブルをバックアップする方法、前回のバックアップ以降の変更内容だけをバックアップする方法、バックアップデータを効率的に別のサーバーに転送する方法など、[mysqlbackup](#) コマンドの機能と構文について説明しています。

それぞれのリリースでの変更点を詳細に記したノートについては、[MySQL Enterprise Backup 3.11 リリースノート](#)を参照してください。

法的情報については、[法的通知](#)を参照してください。

ドキュメント生成日: 2020-11-09 (revision: 79)

目次

序文と法的通知	ix
I MySQL Enterprise Backup の使用開始	1
1 MySQL Enterprise Backup の概要	5
1.1 バックアップのタイプ	5
1.2 mysqlbackup コマンド	6
1.3 バックアップのパフォーマンスおよび容量に関する考慮事項の概要	6
1.4 バックアップされるファイル	7
1.5 データベースのリストアの概要	15
2 MySQL Enterprise Backup のインストール	17
II MySQL Enterprise Backup の使用	19
3 データベースサーバーのバックアップ	23
3.1 最初のバックアップの前に	23
3.1.1 データベース情報を収集する	23
3.1.2 MySQL の権限をバックアップ管理者に与える	25
3.1.3 バックアップデータの場所を指定する	25
3.2 通常のバックアップ/検証/リストアのサイクル	26
3.2.1 MySQL インスタンス全体のバックアップ	26
3.2.2 バックアップの検証	28
3.2.3 元の場所でのデータベースのリストア	28
3.3 バックアップのシナリオと例	29
3.3.1 完全バックアップの作成	29
3.3.2 増分バックアップの作成	30
3.3.3 圧縮バックアップの作成	32
3.3.4 部分バックアップの作成	33
3.3.5 単一ファイルバックアップの作成	37
3.3.6 オプティミスティックバックアップの作成	40
3.3.7 インメモリーデータベースデータのバックアップの作成	41
3.3.8 スケジュールされたバックアップの作成	42
4 データベースのリカバリとリストア	43
4.1 バックアップをリストアする準備	43
4.2 リストア操作の実行	44
4.3 ホットバックアップからのポイントインタイム・リカバリ	46
4.4 単一の .ibd ファイルのバックアップとリストア	47
4.5 データベースのアップグレードまたはダウングレードによるバックアップのリストア	48
5 mysqlbackup コマンドリファレンス	51
5.1 mysqlbackup コマンド行オプション	52
5.1.1 サブコマンド	57
5.1.2 標準オプション	63
5.1.3 接続オプション	64
5.1.4 サーバリポジトリオプション	65
5.1.5 バックアップリポジトリオプション	67
5.1.6 メタデータオプション	70
5.1.7 圧縮オプション	70
5.1.8 増分バックアップオプション	71
5.1.9 部分バックアップとリストアオプション	74
5.1.10 単一ファイルバックアップオプション	80
5.1.11 パフォーマンス/スケーラビリティ/容量オプション	82
5.1.12 メッセージロギングオプション	88
5.1.13 進行状況レポートオプション	89
5.1.14 暗号化オプション	92
5.1.15 クラウドストレージオプション	93
5.1.16 特別なバックアップの種類オプション	94
5.2 構成ファイルとパラメータ	95
6 レプリケーションで MySQL Enterprise Backup を使用する	97
6.1 新しいレプリケーションスレーブのセットアップ	97
6.2 スレーブデータベースのバックアップとリストア	98
6.3 マスターデータベースのリストア	98

7 MySQL Enterprise Backup のパフォーマンスに関する考慮事項	101
7.1 バックアップパフォーマンスの最適化	101
7.2 リストアパフォーマンスの最適化	104
8 バックアップの暗号化	107
9 MySQL Enterprise Backup をメディア管理ソフトウェア (MMS) 製品と使用する	109
9.1 Oracle Secure Backup によるテープへのバックアップ	109
10 MySQL Enterprise Backup のトラブルシューティング	111
10.1 MySQL Enterprise Monitor によるバックアップのモニタリング	111
10.2 MySQL Enterprise Backup のエラーコード	111
10.3 破損の問題の回避	111
10.4 MySQL Enterprise Backup ログの使用	112
10.5 MySQL Enterprise Backup マニフェストの使用	114
11 MySQL Enterprise Backup のよくある質問	115
III 付録	117
A MySQL Enterprise Backup の制限	121
A.1 MySQL Enterprise Backup の制限	121
B MySQL Enterprise Backup の互換性情報	123
B.1 プラットフォーム間の互換性	123
B.2 古い MySQL Enterprise Backup との互換性	123
B.3 古い MySQL バージョンとの互換性	123
B.4 特定の MySQL バージョンに関する互換性の注意事項	123
C 詳細な例	125
C.1 完全バックアップのディレクトリ構造例	125
C.2 圧縮バックアップのディレクトリ構造例	129
C.3 増分バックアップのディレクトリ構造例	129
D MySQL Enterprise Backup リリースノート	131
E サードパーティーコンポーネントライセンス	133
E.1 RegEX-Spencer ライブラリライセンス	133
E.2 zlib ライセンス	133
E.3 LZ4 ライセンス	134
E.4 Percona Multiple I/O スレッドパッチライセンス	134
E.5 Google SMP Patch ライセンス	135
E.6 Google Controlling Master Thread I/O Rate Patch ライセンス	135
E.7 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) ライセンス	136
E.8 OpenSSL v1.0 ライセンス	136
E.9 cURL (libcurl) ライセンス	138
MySQL Enterprise Backup 用語集	139
索引	151

表の一覧

1.1 MySQL Enterprise Backup 出力ディレクトリ内のファイル	7
3.1 データベースのバックアップに必要な情報	23
5.1 すべてのオプションのリスト	53

例の一覧

3.1 InnoDB テーブルの非圧縮部分バックアップの作成	36
3.2 圧縮部分バックアップの作成	36
3.3 絶対パスへの単一ファイルバックアップ	37
3.4 相対パスへの単一ファイルバックアップ	37
3.5 標準出力への単一ファイルバックアップ	37
3.6 単一イメージへの既存のバックアップディレクトリの変換	38
3.7 バックアップディレクトリへの既存のイメージの抽出	38
3.8 単一ファイルバックアップの内容の一覧表示	38
3.9 現在のディレクトリへの単一ファイルバックアップの抽出	38
3.10 バックアップディレクトリへの単一ファイルバックアップの抽出	38
3.11 単一ファイルの選択的な抽出	38
3.12 単一ディレクトリの選択的な抽出	38
3.13 絶対パス名の処理	38
3.14 リモートホストへの単一ファイルバックアップ	39
3.15 リモートの MySQL Server への単一ファイルバックアップ	39
3.16 クラウドバックアップの作成	40
3.17 クラウドストレージからバックアップディレクトリへの既存のイメージの抽出	40
3.18 オプション <code>optimistic-time=YYMMDDHHMMSS</code> を使用したオプティミスティックバックアップ	41
3.19 オプション <code>optimistic-time=now</code> を使用したオプティミスティックバックアップ	41
3.20 <code>optimistic-busy-tables</code> オプションを使用したオプティミスティックバックアップ	41
3.21 <code>optimistic-busy-tables</code> オプションと <code>optimistic-time</code> オプションの両方を使用したオプティミスティックおよび部分バックアップ	41
4.1 バックアップへのログの適用	44
4.2 圧縮バックアップへのログの適用	44
4.3 完全バックアップへの増分バックアップの適用	44
4.4 データベースのシャットダウンとリストア	44
4.5 圧縮バックアップのリストア	45
4.6 暗号化されたバックアップイメージのリストア	45
4.7 増分バックアップイメージのリストア	45
4.8 TTS バックアップからの選択されたテーブルのリストア	46
4.9 クラウドストレージから MySQL Server への単一ファイルバックアップのリストア	46
5.1 デフォルトの構成ファイルからの接続パラメータによる簡単なバックアップ	58
5.2 基本的な増分バックアップ	58
5.3 フルバックアップへのログの適用	58
5.4 増分バックアップ	74
5.5 <code>backup-my.cnf</code> ファイルの例	95
9.1 Oracle Secure Backup と MySQL Enterprise Backup を使用したサンプル <code>mysqlbackup</code> コマンド	110

序文と法的通知

本書は MySQL Enterprise Backup 製品のユーザーマニュアルです。

ライセンス情報については、[法的通知](#)を参照してください。この製品には、サードパーティーのコードが含まれる場合があります。サードパーティーコードのライセンス情報については、[付録E「サードパーティーコンポーネントライセンス」](#)を参照してください。

法律上の注意点

Copyright © 2003, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アSEMBル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション (人的傷害を発生させる可能性があるアプリケーションを含む) への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。MySQL は Oracle Corporation およびその関連会社の商標であり、Oracle による明確な書面による認可なしに使用できません。その他の社名、商品名等は各社の商標または登録商標である場合があります。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様と Oracle Corporation との間の契約に別段の定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様と Oracle Corporation との間の契約に定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

このドキュメントには、ソフトウェアまたは印刷物などの形式を問わず、オラクルが独占的な権利を有する財産的情報が含まれています。この機密資料へのアクセスと使用は、お客様とオラクルとの間で締結され、お客様が遵守に同意した、Oracle Master Agreement、Oracle License and Oracle Services Agreement(オラクル・ライセンスおよびサービスに関する契約書)、Oracle PartnerNetwork Agreement、Oracle Distribution Agreementまたはその他のライセンス契約の条件に従うものとし、このドキュメントとその内容の開示、コピー、複製および配布には、オラクルによる事前の承諾を必要とします。このドキュメントはライセンス契約の一部となるものではなく、オラクルおよびその子会社や関連会社との契約を構成するものではありません。

このドキュメントは、GPL ライセンスに基づき配布されるものではありません。このドキュメントの使用は、次の条項に従います。

個人的な使用にかぎり、このドキュメントの出力コピーを作成できます。実際のコンテンツをいっさい変更したり編集したりしないかぎり、ほかの形式に変換することは許可されています。このドキュメントはいかなる形態またはいかなるメディアであっても公開または配布されないものとします。ただし、ドキュメントが同じメディアでソフトウェアと一緒に配布されるという条件で、Oracle がこのドキュメントを配布する方法と類似した方法で (すなわち、このソフトウェアを掲載する Web サイトから電子的にダウンロードする場合)、または CD-ROM や類似のメディアでドキュメントを配布する場合を除きます。出力コピーを配布するなど、ほかの形式で使用したり、別の出版物でこのドキュメントの全部または一部を使用したりする場合は、権限を有する Oracle の代表者から事前の書面による同意を得る必要があります。Oracle およびその関連会社は、上記で特に認められている場合を除き、このドキュメントに対するあらゆる権利を保有します。

このライセンスの条項または MySQL ドキュメントの作成方法に関する詳細は、[MySQL Contact & Questions](#) にアクセスしてください。

MySQL 製品が使用するサードパーティーライブラリのライセンスを含め、ライセンスに関する追加情報については、[序文と法的通知](#)を参照してください。

MySQL の使用に関するヘルプは、抱えている問題についてほかの MySQL ユーザーに相談できる [MySQL フォーラム](#)または [MySQL メーリングリスト](#)にアクセスしてください。

ほかの言語へのドキュメントの翻訳を含む MySQL 製品の追加情報と、HTML 形式や PDF 形式などのさまざまな形式でダウンロードできるバージョンについては、[MySQL Documentation Library](#) を参照してください。

第 I 部 MySQL Enterprise Backup の使用開始

目次

1 MySQL Enterprise Backup の概要	5
1.1 バックアップのタイプ	5
1.2 mysqlbackup コマンド	6
1.3 バックアップのパフォーマンスおよび容量に関する考慮事項の概要	6
1.4 バックアップされるファイル	7
1.5 データベースのリストアの概要	15
2 MySQL Enterprise Backup のインストール	17

第 1 章 MySQL Enterprise Backup の概要

目次

1.1 バックアップのタイプ	5
1.2 mysqlbackup コマンド	6
1.3 バックアップのパフォーマンスおよび容量に関する考慮事項の概要	6
1.4 バックアップされるファイル	7
1.5 データベースのリストアの概要	15

MySQL Enterprise Backup 製品は、MySQL データのバックアップ操作を実行します。あらゆる種類の MySQL テーブルをバックアップできます。これは、InnoDB テーブルの高速で便利なバックアップに対して特別に最適化されています。InnoDB バックアップが高速で InnoDB テーブルの信頼性と拡張性が高いため、もっとも重要なデータには InnoDB テーブルを使用することをお勧めします。

この本では、MySQL のバックアップに関するベストプラクティスについて説明し、MySQL Enterprise Backup 機能を使用してこれらのプラクティスを実装する方法を記載しています。この本では次の内容について説明します。

- バックアップが重要である理由。
- MySQL データベースを構成するファイルとそれらの果たす役割。
- バックアップ中にデータベースの実行を継続させる方法。
- バックアップジョブの時間、CPU オーバーヘッド、およびストレージオーバーヘッドを最小化する方法。多くの場合、これらのいずれかの側面を最小にすると別の側面が増大します。
- 災害が発生した場合のデータのリストア方法。バックアップを検証しリカバリを実施する方法について学習し、プレッシャーがある場合でも平静と自信を保てるようになります。
- 日常的な管理と新しいサーバーの配備でバックアップデータを使用するほかの方法。

1.1 バックアップのタイプ

さまざまな種類のバックアップ手法は、ホット (もっとも望ましい) からコールド (もっとも中断が多い) までの段階で分類されます。バックアップの進行中でも継続して、データベースシステムと、関連したアプリケーションおよび Web サイトを動作および応答できるようにすることが目標です。

ホットバックアップは、データベースが実行している間に実行されます。このタイプのバックアップは、通常のデータベース操作を妨げません。バックアップが行われている間に生じた変更内容も取得します。これらの理由により、データベースが「増大」する次の場合にはホットバックアップが最適です。バックアップに非常に長い時間がかかるほどデータが大きな場合や、アプリケーション、Web サイト、または Web サービスをオフラインにすることなく、ビジネスにとって非常に重要なデータの最新の変更内容をすべて取得する必要がある場合。

MySQL Enterprise Backup は、すべての InnoDB テーブルのホットバックアップを行います。MyISAM テーブルとほかの非 InnoDB テーブルは、**ウォームバックアップ**手法を使用して最後にバックアップされます。データベースは実行し続けますが、バックアップフェーズ中にシステムは読み取り専用状態になります。

データベースが停止している間に**コールドバックアップ**を実行することもできます。サービスの停止を回避するには、通常、このようなバックアップは、アプリケーション全体または Web サイトをダウンさせずに停止できるレプリケーションスレーブから実行します。

留意点

ホットバックアップフェーズ中にできるだけ多くのデータをバックアップするには、新しいテーブルに対するデフォルトのストレージエンジンとして InnoDB を指定するか、InnoDB ストレージエンジンを使用するように既存のテーブルを変換します。(MySQL 5.5 以上では、現在 InnoDB が新しいテーブルのデフォルトのストレージエンジンになっています。)

ホットバックアップとウォームバックアップ中、データベースの構造に関する情報は、自動的にデータベース接続を通じて取得されます。コールドバックアップの場合、構成ファイルまたはコマンド行オプションを通じてファイルの場所を指定する必要があります。

1.2 mysqlbackup コマンド

MySQL Enterprise Backup 製品を使用する場合、主に `mysqlbackup` コマンドを使用します。このコマンドは、指定したオプションに基づいて、さまざまなタイプのバックアップ操作およびリストア操作もすべて実行します。`mysqlbackup` は、バックアップごとにタイムスタンプが記されたサブディレクトリの作成、バックアップデータの圧縮、別のサーバーへの転送を簡単にするためのバックアップデータの単一ファイルへのパッキングなど、このコマンドを使用しなければ自身のバックアップスクリプトにコード化していたほかの作業も行えます。

`mysqlbackup` 機能の使用方法については、第3章「データベースサーバーのバックアップ」を参照してください。オプションの構文については、第5章「`mysqlbackup` コマンドリファレンス」を参照してください。

1.3 バックアップのパフォーマンスおよび容量に関する考慮事項の概要

バックアップ戦略では、パフォーマンスとストレージ領域が重要な側面になります。ユーザーは、データベースサーバーでの CPU オーバーヘッドを小さくして、バックアップを迅速に完了させることを希望しています。また、バックアップデータをコンパクトにして、即座にリストアできるように複数のバックアップを手元に用意しておくことも希望しています。別のシステムへのバックアップデータの転送は、迅速で便利に行えることが必要です。このようなすべての側面は、`mysqlbackup` コマンドのオプションによって制御されます。

場合によっては、さまざまな種類のオーバーヘッド (CPU サイクル、ストレージ領域、ネットワークトラフィック) のバランスをとる必要があります。計画メンテナンス中、または災害が発生したときにデータのリストアに要する時間を必ず意識してください。たとえば、MySQL Enterprise Backup の主要ないくつかの機能について、次のような検討要素があります。

- **並列バックアップ** は、MySQL Enterprise Backup 3.8 でデフォルトであり、以前の MySQL Enterprise Backup リリースに対してパフォーマンスを大きく改善しています。読み取り、処理、および書き込みが、すべての MEB 操作の主な下位操作になります。たとえば、バックアップ操作で、MySQL Enterprise Backup は最初に、ディスクからデータを読み取り、続いてこのデータを処理し、データをディスクに書き込み、再度データを読み取って検証します。MySQL Enterprise Backup では、これらの下位操作が互いに独立しており、並列して実行できるため、パフォーマンスが向上します。読み取り、処理、および書き込みの下位操作は、同じ種類の複数のスレッド (複数の読み取りスレッド、複数の処理スレッド、および複数の書き込みスレッド) を使用して並列的に実行され、その結果パフォーマンスが向上します。パフォーマンスの向上は通常、ソースデバイスとターゲットデバイスの両方で RAID アレイが使用される場合と、より多くの CPU サイクルを並列して使用できる圧縮バックアップの場合に大きくなります。

並列バックアップは、16M バイトのブロックを使用したブロックレベルの並列処理を採用しています。異なるスレッドは、単一ファイル内の別々の 16M バイトのチャンクに対して読み取り、処理、および書き込みを行います。並列バックアップは、インスタンスに単一の大きな **システムテーブルスペース** が含まれているか、(`innodb_file_per_table` モードで作成された `.ibd` ファイルによって表される) 小さな多数のテーブルスペースが含まれているかにかかわらず、操作のパフォーマンスを改善します。

- **増分バックアップ** は完全バックアップより高速で、データベースサーバー上のストレージ領域を節約し、別のサーバーにバックアップデータを転送するネットワークトラフィックを軽減します。増分バックアップでは、リストアできるようにバックアップを準備しておく追加処理が必要になります。この処理は別のシステムで実行できるため、データベースサーバーでの CPU オーバーヘッドを最小限に抑えることができます。
- **圧縮バックアップ** は InnoDB テーブル用のストレージ領域を節約し、バックアップデータを別のサーバーに転送するネットワークトラフィックを軽減します。このバックアップでは、非圧縮バックアップより多くの CPU オーバーヘッドが発生します。リストア中、圧縮されたデータと圧縮解除されたデータが同時に必要になるため、この追加ストレージ領域とデータを圧縮解除する時間を考慮に入れてください。

InnoDB テーブル内のデータを圧縮する以外に、圧縮バックアップは、InnoDB テーブルスペースファイル内の未使用領域のスキップも行います。非圧縮バックアップにはこの未使用領域が含まれます。

- 領域が限られている場合や、安価で大容量だが速度が遅いテープなどのストレージデバイスを使用している場合は、パフォーマンスと領域に関する考慮事項は異なります。できるだけ高速なバックアップを目的とせず、データベースサーバーにバックアップデータの間コピーを格納しないようにしたい場合があります。MySQL Enterprise

Backup は、単一ファイルのバックアップを生成して、そのファイルをほかのサーバーまたはデバイスに直接ストリーミングすることができます。バックアップデータはローカルシステムに一切保存されないため、データベースサーバーでの領域オーバーヘッドが回避されます。また、一連のバックアップファイルを保存し、続いて別のサーバーまたはストレージデバイスへの転送用にそれらをアーカイブにまとめるというパフォーマンスオーバーヘッドも回避されます。詳細は、[セクション 3.3.5.1 「別のデバイスまたはサーバーへのバックアップデータのストリーミング」](#)を参照してください。

圧縮を行うためのデータベースサーバーでの CPU オーバーヘッドが、テープデバイス上でのストレージ領域の追加よりも高価なため、バックアップデータをテープにストリーミングするときに通常はバックアップを圧縮しません。別のサーバーにバックアップデータをストリーミングするときに、どのサーバーに CPU 能力の余裕が多いか、圧縮によってどれだけネットワークトラフィックを軽減できるかに応じて、元のサーバーで圧縮することも、ストリーミング先のサーバーで圧縮することもできます。または、即座にリストアできるように、バックアップデータをストリーミング先のサーバーで圧縮せずに保存しておくこともできます。

データをリカバリする速度が重要である障害リカバリの場合、重要なバックアップデータを準備して圧縮解除された状態にしておく、リストア操作のステップ数を最小限に抑えることができます。

速度が非常に重要になるのは障害リカバリ中です。たとえば、`mysqldump` コマンドで実行された[論理バックアップ](#)は、MySQL Enterprise Backup 製品を使用した[物理バックアップ](#)とほぼ同じ時間がかかりますが（少なくとも小さなデータベースの場合）、リストア操作は通常、MySQL Enterprise Backup のほうが速くなります。実際のデータファイルをコピーしてデータディレクトリに戻すと、`mysqldump` 出力から SQL ステートメントを再実行して生じる、行を挿入しインデックスを更新するというオーバーヘッドはスキップされます。

Linux システムと Unix システムでのサーバーパフォーマンスへのあらゆる影響を最小限に抑えるために、MySQL Enterprise Backup は、`posix_fadvise()` システムコールを使用することによって、オペレーティングシステムのディスクキャッシュに格納せずにバックアップデータを書き込みます。この手法は、バックアップデータを大量に 1 回限り読み取る操作で、頻繁にアクセスされるデータがディスクキャッシュからフラッシュされないようにすることにより、バックアップ操作後の速度低下を最小限に抑えます。

手法の詳細と、バックアップおよびリストアのパフォーマンスにかかわるトレードオフの詳細は、[第 7 章 「MySQL Enterprise Backup のパフォーマンスに関する考慮事項」](#)を参照してください。

1.4 バックアップされるファイル

DBA と開発作業には通常、テーブル、行、カラム、データディクショナリなどの論理構造が含まれます。バックアップでは、これらの構造がファイルによってどのように表されているかに関する物理的な詳細を理解しておく必要があります。

表 1.1 MySQL Enterprise Backup 出力ディレクトリ内のファイル

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
<code>ibdata*</code>	InnoDB システムテーブルスペース。複数の InnoDB テーブルおよび関連したインデックスが含まれます。	バックアップの進行中に元のファイルが変更される可能性があるため、 <code>apply-log</code> ステップで、対応するバックアップファイルに同じ変更内容が適用されます。
<code>*.ibd</code>	InnoDB file-per-table テーブルスペース。それぞれに単一の InnoDB テーブルおよび関連したインデックスが含まれます。	<code>innodb_file_per_table</code> 下で作成されたテーブルに使用されます。バックアップの進行中に元のファイルが変更される可能性があるため、 <code>apply-log</code> ステップで、対応するバックアップファイルに同じ変更内容が適用されます。
<code>*.ibz</code>	MySQL データディレクトリからの InnoDB データファイルの圧縮された形式。	圧縮バックアップで <code>.ibd</code> ファイルの代わりに生成されます。InnoDB システムテーブルスペースを表す <code>ibdata*</code> ファイルも、圧縮バックアップではこの拡張子になります。 <code>.ibz</code> ファイルは、 <code>apply-log</code> ステップのために圧縮解除されます。

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
*.frm	すべての MySQL テーブルに関するメタデータを保持します。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.MYD	MyISAM テーブルデータ。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.MYI	MyISAM インデックスデータ。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.CSM	CSV テーブルのメタデータ。	これらのファイルは変更なしにコピーされます。mysqlbackup で作成される backup_history テーブルと backup_progress テーブルでは CSV 形式が使用されるため、バックアップには常にこの拡張子のファイルが含まれます。
*.CSV	CSV テーブルのデータ。	これらのファイルは変更なしにコピーされます。mysqlbackup で作成される backup_history テーブルと backup_progress テーブルでは CSV 形式が使用されるため、バックアップには常にこの拡張子のファイルが含まれます。
*.MRG	MERGE ストレージエンジンはほかのテーブルを参照します。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.TRG	トリガーパラメータ。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.TRN	トリガー名前空間情報。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.opt	データベース構成情報。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.par	パーティション化されたテーブルの定義。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.ARM	アーカイブストレージエンジンメタデータ。	これらのファイルがコピーされている間、データベースは読み取り専用状態になります。これらのファイルは変更なしにコピーされます。
*.ARZ	アーカイブストレージエンジンデータ。	これらのファイルがコピーされている間、データベースは読み取り専用状態

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
		になります。これらのファイルは変更なしにコピーされます。
backup-my.cnf	MySQL データファイルのレイアウトを指定する構成パラメータを記録します。	リストア操作で、バックアップが作成された時点と同じレイアウトを再生成するために使用されます。
ibbackup_logfile	MySQL データディレクトリからの ib_logfile* ファイルの短縮バージョン。	InnoDB ログファイル (ib_logfile*) は、データベース操作中に継続的に更新される固定サイズのファイルです。バックアップの目的では、バックアップが進行している間にコミットされた変更内容だけが必要になります。これらの変更内容は、 ibbackup_logfile に記録され、apply-log フェーズ中に ib_logfile* ファイルの再作成に使用されます。
ibbackup_redo_log_only	<code>--incremental-with-redo-log-only</code> オプションを使用して行われる増分バックアップで、 ibbackup_logfile の代わりに使用されます。	
ib_logfile*	最初のバックアップ後、apply-log フェーズ中にバックアップディレクトリに作成されます。	これらのファイルは元のデータディレクトリからコピーされるのではなく、 ibbackup_logfile ファイルに記録された変更内容を使用して、最初のバックアップ後、apply-log フェーズ中にバックアップディレクトリに再作成されます。
*.bl	バックアップ対象サーバーからの各 .isl ファイルの名前を変更したバージョン。	<code>CREATE TABLE ... DATA DIRECTORY = ...</code> の構文を使用して InnoDB テーブルの場所を指定するときに、 .isl ファイルは作成され、テーブルスペースファイルを指し示すシンボリックリンクのように機能します。(詳細は テーブルスペースの位置の指定 を参照してください。) .bl ファイルは、 <code>copy-back</code> 操作中に .isl ファイルに戻される場合も戻されない場合もあります。バックアップがリストアされるサーバーに、指定されたディレクトリが存在しない場合、 <code>mysqlbackup</code> コマンドはこのディレクトリを作成しようと試みます。ディレクトリを作成できない場合、リストア操作は失敗します。したがって、 <code>DATA DIRECTORY</code> 句を使用してテーブルを異なる場所に配置し、対応するディレクトリを作成できずファイル構造が異なるサーバーにリストアする場合は、リストアする前に、リストア先のサーバー上に存在するディレクトリを指し示すように .bl ファイルを編集してください。
2011-05-26_13-42-02 などのタイムスタンプが記されたディレクトリ	<code>--with-timestamp</code> オプションで作成されます。すべてのバックアップファイルはこのサブディレクトリ内に保存されます。	複数セットのバックアップデータを同一の主要なバックアップディレクトリ下で継続的に利用する場合は必ず、 <code>--</code>

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
		<p><code>with-timestamp</code> オプションを使用してください。</p>
<p><code>datadir</code> ディレクトリ</p>	<p>元の MySQL インスタンスからのすべてのデータファイルとデータベースサブディレクトリを格納するサブディレクトリ。</p>	<p><code>mysqlbackup</code> コマンドによって、バックアップディレクトリ下に作成されます。</p>
<p>バイナリログファイル</p>	<p>サーバーからのバイナリログファイル。これらはデフォルトでバックアップに含まれます。これにより、サーバーのスナップショットを取得できるため、サーバーを正確にクローン化できます。完全バックアップを基本として使用しながら、増分バックアップに含まれるバイナリログファイルは、前回の完全バックアップ以降の特定の時点の状態にデータベースをリストアするポイントインタイム・リカバリ (PITR) に使用できます。詳細は、セクション4.3「ホットバックアップからのポイントインタイム・リカバリ」を参照してください。</p>	<p>バックアップディレクトリ下の <code>datadir</code> ディレクトリの下に保存されます。<code>--skip-binlog</code> オプションを使用して、バックアップでバイナリログを除外します。MySQL 5.5 の場合はすべてのオフラインバックアップと同様に、<code>mysqlbackup</code> がバイナリログファイルを見つけてバックアップに含めることができるように、<code>--log-bin-index</code> オプションを使用して、使用中のすべてのバイナリログファイルを一覧表示した、MySQL サーバー上のインデックスファイルの絶対パスを指定してください (オプションのデフォルト値と異なる場合)。</p> <p>注記</p> <p>一部の既知の問題のため、オフラインバックアップを作成するときや、<code>--no-locking</code> オプションで作成された完全バックアップに基づいた増分バックアップを作成するときには、ユーザーは常に <code>--skip-binlog</code> オプションを使用する必要があります。詳細は、付録 A「MySQL Enterprise Backup の制限」を参照してください。</p>
<p>リレーログファイル</p>	<p>スレーブサーバーからのリレーログファイル。これらは、デフォルトでスレーブサーバーのバックアップに含</p>	<p>バックアップディレクトリ下の <code>datadir</code> ディレクトリの下に保存されます。バックアップでリレーログを除</p>

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
	められます。これらが含まれているため、スレーブがリストアされているときに、マスターからリレーログを取り出すために必要な時間とリソースが節約されます。	外するには、 <code>--skip-relaylog</code> オプションを使用してください。オフラインバックアップの場合、 <code>mysqlbackup</code> がリレーログファイルを見つけてバックアップに含めることができるように、 <code>--relay-log-index</code> オプションを使用して、使用中のすべてのリレーログファイルを一覧表示した、MySQL サーバー上のインデックスファイルの絶対パスを指定してください (オプションのデフォルト値と異なる場合)。
スレーブステータスログファイル	通常、これらのファイルは <code>master.info</code> および <code>relay-log.info</code> という名前が付けられ、デフォルトで、レプリケーションを行うときにスレーブデータベースのバックアップに含められます。詳細は、 スレーブステータスログ を参照してください。	バックアップディレクトリ下の <code>datadir</code> ディレクトリの下に保存されます。オフラインバックアップの場合、 <code>mysqlbackup</code> がこれらのファイルを見つけてバックアップに含めることができるように、 <code>--master-info-file</code> オプションと <code>--relaylog-info-file</code> オプションを使用して、情報ファイルの絶対パスを指定してください (オプションのデフォルト値と異なる場合)。
画像ファイル	<code>--backup-image</code> オプションで指定された名前が付けられた、 <code>backup-to-image</code> オプションによって生成された単一ファイルのバックアップ。	バックアップデータディレクトリがゼロバイトのファイルのみで構成され、最上位のディレクトリに単一の巨大なデータファイルがある場合は、単一ファイルのバックアップが存在しています。内部のコンテンツを損失または損傷することなくイメージファイルを移動し、続いて <code>extract</code> オプションを使用し、 <code>--backup-image</code> オプションで同じイメージ名を指定した <code>mysqlbackup</code> コマンドでそのファイルをアンパックします。 <code>backup-my.cnf</code> などの一部の付加的なファイルと <code>meta</code> サブディレクトリはバックアップディレクトリに存在しますが、これらのファイルはイメージファイルにも含められ、一緒に移動する必要はありません。
ほかのすべてのファイル	MySQL データディレクトリからコピーされます。	デフォルトで、MySQL データディレクトリ内の認識されないファイルはすべて、バックアップにコピーされます。このようなファイルを除外するには、 <code>--only-known-file-types</code> オプションを指定してください。
meta ディレクトリ	バックアップに関するメタデータを含むファイルを格納したサブディレクトリ。	<code>mysqlbackup</code> コマンドによって、バックアップディレクトリ下に作成されます。次に一覧表示されたすべてのファイルは、 <code>meta</code> サブディレクトリ内に置かれます。
<code>backup_variables.txt</code>	バックアップに関する重要な情報が保持されます。 <code>mysqlbackup</code> コマンド専用です。MySQL Enterprise Backup 3.6 以前では、この情報は	<code>mysqlbackup</code> コマンドは、 <code>apply-log</code> フェーズや <code>restore</code> フェーズなど、最初のバックアップ後の操作中にこのファイルを参照し、多くの場合更新します。

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
	<code>ibbackup_binlog_info</code> という名前のファイルに含まれていました。	
<code>image_files.xml</code>	<code>backup-to-image</code> オプションまたは <code>backup-dir-to-image</code> オプションで生成された単一ファイルのバックアップに存在するすべてのファイル (それ自身を除く) のリストが含まれます。このファイルの詳細は、 セクション 10.5 「MySQL Enterprise Backup マニフェストの使用」 を参照してください。	このファイルはいったん生成されると、どの段階でも変更されません。
<code>backup_create.xml</code>	コマンド行引数と、バックアップが作成された環境を一覧表示します。このファイルの詳細は、 セクション 10.5 「MySQL Enterprise Backup マニフェストの使用」 を参照してください。	このファイルはいったん作成されると、変更されません。 <code>--disable-manifest</code> オプションを指定することによって、このファイルが生成されないようにすることができます。
<code>backup_content.xml</code>	バックアップデータのファイルおよびデータベース定義の基本メタデータ。バックアップ対象サーバー上で定義されているすべてのプラグインの詳細も含まれます。ユーザーはこの詳細から、同じプラグインがリストアのターゲットサーバー上でも同じように定義されていることを確認する必要があります。このファイルの詳細は、 セクション 10.5 「MySQL Enterprise Backup マニフェストの使用」 を参照してください。	このファイルはいったん作成されると、変更されません。 <code>--disable-manifest</code> オプションを指定することによって、このファイルが生成されないようにすることができます。
<code>comments.txt</code>	<code>--comments</code> オプションまたは <code>--comments-file</code> オプションによって生成されます。	コメントは、このバックアップジョブの目的または特別な考慮事項について記すためにユーザーが指定します。
<code>gtid_executed.sql</code>	GTID を有効にしたサーバーから取得されたバックアップを指定します。	GTID は、MySQL 5.6 以上のレプリケーション機能です。詳細は、 グローバルトランザクション識別子を使用したレプリケーション を参照してください。GTID を有効にしたサーバーをバックアップすると、バックアップディレクトリに <code>gtid_executed.sql</code> のファイルが作成されます。スレーブサーバー上にバックアップデータをリストアしたあとで、このファイルを編集し実行します。
<code>server-my.cnf</code>	デフォルト以外の値に設定された、バックアップ対象サーバーのグローバル変数の値が含まれます。このファイルまたは <code>server-all.cnf</code> を使用して、リストアのターゲットサーバーを起動します。	<code>copy-back</code> または <code>copy-back-and-apply-log</code> 操作中、ファイル内の <code>サーバーリポジトリオプション</code> の値 (<code>--datadir</code> 、 <code>--innodb_data_home_dir</code> など) は、コマンドオプションを通じてコマンドによって変更がなされた場合に変更されます。ただし、 <code>--apply-incremental-backup</code> 操作中は、ファイルにすでに保存された値が優先され、コマンドを通じて与えられるオプション値によって変更されません。

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
		<p>警告</p> <p>ファイルを使用してターゲットサーバーを再起動する際には、ターゲットサーバーが誤って間違ったファイルの場所を使用しないようにするために、<code>--tmpdir</code>、<code>--general-log</code>などのパラメータと、絶対ファイルパスを使用するすべてのグローバル変数を変更してください。</p>
<p><code>server-all.cnf</code></p>	<p>バックアップ対象サーバーのすべてのグローバル変数の値が含まれます。このファイルまたは <code>server-my.cnf</code> を使用して、リストアのターゲットサーバーを起動します。</p>	<p><code>copy-back</code> または <code>copy-back-and-apply-log</code> 操作中、ファイル内のサーバーリポジトリオプションの値 (<code>--datadir</code>、<code>--innodb_data_home_dir</code> など) は、コマンドオプションを通じてコマンドによって変更がなされた場合に変更されます。ただし、<code>--apply-incremental-backup</code> 操作中は、ファイルにすでに保存された値が優先され、コマンドを通じて与えられるオプション値によって変更されません。</p> <p>警告</p> <p>ファイルを使用してターゲットサーバーを再起動する際には、ターゲットサーバーが誤って間違ったファイルの場所を使用しないようにするために、<code>--tmpdir</code>、<code>--general-log</code>などのパラ</p>

ファイル名、パターン、または拡張子	元のデータファイルとの関係	注
		メータと、絶対ファイルパスを使用するすべてのグローバル変数を変更してください。

InnoDB データ

InnoDB ストレージエンジンで管理されるデータは常にバックアップされます。バックアップされる主な InnoDB 関連データには、[システムテーブルスペース](#)とおそらくはいくつかのユーザーテーブルのデータを表す `ibdata*` ファイル、[file-per-table](#) 設定を有効にして作成されたユーザーテーブルからのデータを含む `.ibd` ファイル、新しいバックアップファイル `ibbackup_logfile` に格納される `ib_logfile*` ファイル (バックアップの実行中に行われた変更内容を表す再実行ログ情報) から抽出されたデータなどがあります。

圧縮バックアップ機能を使用する場合、`.ibd` ファイルは、圧縮形式で `.ibz` ファイル に名前が変更されます。

これらのファイルは、最初にコピーされるときに、リストアする準備が整うまでにさらなる処理を必要とする [raw](#) バックアップを形成します。続いて、[適用](#)ステップを実行すると、`ibbackup_logfile` ファイルに記録された変更内容に基づいてバックアップファイルが更新され、[準備されたバックアップ](#)が作成されます。この時点で、バックアップデータは単一時点に対応します。ファイルはこれで、元の場所にリストアしたり、テスト、レポート、配備などのほかのいくつかの用途のためにレプリケーションスレーブとしてリストアしたりする準備ができました。

InnoDB テーブルを元の状態にリストアするには、バックアップデータとともに対応する `.frm` ファイルも必要です。このファイルがないと、バックアップ以後にだれかが `ALTER TABLE` または `DROP TABLE` ステートメントを実行した場合には、テーブル定義が欠落または失効する可能性があります。デフォルトでは、`mysqlbackup` コマンドは自動的に `.frm` ファイルをバックアップ操作中にコピーし、リストア操作中にファイルをリストアします。

MyISAM およびほかのストレージエンジンからのデータ

`mysqlbackup` コマンドは、MyISAM テーブルの `.MYD` ファイル、`.MYI` ファイル、および関連した `.frm` ファイルもバックアップできます。[このリスト](#)に示されているように、同じことがほかの拡張子のファイルにも適用されます。

注記

MySQL Enterprise Backup では InnoDB 以外のデータ (MyISAM テーブルなど) をバックアップできますが、バックアップする MySQL サーバーで InnoDB をサポートしている必要があります (つまり、サーバーが `--innodb=OFF` または `--skip-innodb` オプションを使用して起動された場合、バックアッププロセスは失敗します)、サーバーに少なくとも 1 つの InnoDB テーブルが含まれている必要があります。

MyISAM テーブルとこれらのほかのタイプのファイルは、InnoDB テーブルで可能な同じ非ブロック方法では、バックアップすることはできません。このフェーズは、[ウォームバックアップ](#)です。これらのテーブルがバックアップされている間は、おそらくはしばらくデータベースが応答しないようにして、これらのテーブルへの変更は妨げられませんが、バックアップ中のシャットダウンは不要です。

注記

処理が集中したデータベースのバックアップ中に起こる同時実行の問題を回避するために、`--only-innodb` オプションまたは `--only-innodb-with-frm` オプションを使用して、InnoDB テーブルと関連データだけをバックアップできます。

バックアップに含まれる生成されたファイル

バックアップデータには、バックアッププロセス中に生成された新しいファイルが含まれます。これらのファイルは、バックアップデータの検証やリストアなど、後から行うタスクを制御するために使用されます。バックアッププロセス中に生成されるファイルは次のとおりです。

- `backup-my.cnf`: バックアップに適用される重大な構成パラメータを記録します。これらのパラメータ値はリストア操作中に使用され、その結果、その間に `my.cnf` ファイルに変更があったかどうかとは無関係に元の値が使用されません。
- `meta/backup_create.xml`: コマンド行引数と、バックアップが作成された環境を一覧表示します。
- `meta/backup_content.xml`: バックアップデータのファイルおよびデータベース定義の基本メタデータ。
- `server-my.cnf`: デフォルト以外の値に設定された、バックアップ対象サーバーのグローバル変数の値が含まれます。
- `server-all.cnf`: バックアップ対象サーバーのすべてのグローバル変数の値が含まれます。

バックアップディレクトリ内のすべてのファイルの詳細は、[表1.1「MySQL Enterprise Backup 出力ディレクトリ内のファイル」](#)を参照してください。

単一ファイルバックアップ

ワークフローによっては、元のインスタンス内のファイルごとに個別のファイルを生成する一般的なバックアップではなく、単一ファイルのバックアップを実行できます。単一ファイル形式のほうが簡単に、別のシステムへの転送や圧縮と圧縮解除を行え、バックアップ対象ファイルが後から誤って削除されないようにすることができます。完全リストアの実行は、複数ファイルのバックアップと同じくらい高速です。個々のファイルのリストアは、複数ファイルのバックアップより遅くなる可能性があります。手順については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

1.5 データベースのリストアの概要

リストアプロセスを開始するには、`copy-back` または `copy-back-and-apply-log` サブコマンドを付けて `mysqlbackup` コマンドを実行します。MySQL サーバーのすべてのデータ、つまりそれぞれに複数のテーブルが含まれている複数のデータベースをリストアできます。または、選択したデータベース、テーブル、あるいはその両方をリストアできます。

データの破損などの問題を修復するには、データを元のサーバーマシンの元の場所にリストアします。別のサーバーマシンまたは別の場所にリストアして、マスターサーバーからのデータで新しいレプリケーションスレーブを設定したり、レポートのためにデータベースをクローニングしたりすることができます。

データベースのリストアの手順については、[第4章「データベースのリカバリとリストア」](#)を参照してください。

第 2 章 MySQL Enterprise Backup のインストール

バックアップする予定のコンテンツを含むそれぞれのデータベースサーバーに MySQL Enterprise Backup 製品をインストールします。MySQL インスタンスと同じサーバーで `mysqlbackup` コマンドを実行することにより、すべてのバックアップおよびリストア操作をローカルで実行します。

オプション: `apply-log` オプションを付けて `mysqlbackup` を実行する場合にのみ、データベースサーバー以外のコンピュータにも MySQL Enterprise Backup 製品をインストールできます。バックアップデータを個別のサーバーに転送し、そのサーバーで「ログの適用」ステップを実行する場合の情報については、[セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」](#) を参照してください。

MySQL Enterprise Backup 製品は、アーカイブファイル (`tar` でアーカイブされ、`gzip` で圧縮された `.tgz`) がプラットフォーム固有のインストーラのどちらかとしてパッケージングされています。

Unix および Linux システムのインストール

すべての Linux および Unix システムで、製品は `.tgz` ファイルとして提供されています。このファイルを次の手順に従ってアンパックします。

```
tar xvzf package.tgz
```

`mysqlbackup` コマンドはサブディレクトリにアンパックされます。システムディレクトリ (実行許可ビットを保存) にこれらをコピーすることも、それをアンパックしたディレクトリを `$PATH` 設定に追加することもできます。

ある特定の Linux ディストリビューションでは、製品は RPM アーカイブとしても提供されています。コマンド `sudo rpm -i package_name.rpm` を使用して RPM をインストールすると、`mysqlbackup` コマンドがディレクトリ `/opt/mysql/meb-3.11` にインストールされます。`$PATH` 設定にこのディレクトリを追加する必要があります。

Windows システムでのインストール

この製品は、MySQL Installer for Windows を使用して、ほかの MySQL 製品と一緒にインストールできます。個々の `.msi` インストーラまたは `.zip` ファイルのどちらかを使用して個別にインストールすることもできます。

`.msi` インストーラでインストールするときは、インストール先を指定します。ほかの MySQL 製品がインストールされている同じディレクトリをお勧めします。「Include directory in Windows PATH」オプションを選択すると、どのディレクトリからでも `mysqlbackup` コマンドを実行できます。

`.zip` ファイルでインストールする場合は、単にファイルを解凍して、`mysqlbackup.exe` を目的のインストール先に保存するだけです。その場所を `%PATH%` 変数に追加すると、どのディレクトリからでも `mysqlbackup` コマンドを実行できます。

「スタート」>「すべてのプログラム」>「MySQL Enterprise Backup 3.11」>「MySQL Enterprise Backup Command Line」のメニュー項目を選択して、インストールできたかどうかを検証します。このメニュー項目によってバージョン情報が表示され、`mysqlbackup` コマンドを実行するためのコマンドプロンプトが開きます。

第 II 部 MySQL Enterprise Backup の使用

目次

3 データベースサーバーのバックアップ	23
3.1 最初のバックアップの前に	23
3.1.1 データベース情報を収集する	23
3.1.2 MySQL の権限をバックアップ管理者に与える	25
3.1.3 バックアップデータの場所を指定する	25
3.2 通常のバックアップ/検証/リストアのサイクル	26
3.2.1 MySQL インスタンス全体のバックアップ	26
3.2.2 バックアップの検証	28
3.2.3 元の場所でのデータベースのリストア	28
3.3 バックアップのシナリオと例	29
3.3.1 完全バックアップの作成	29
3.3.2 増分バックアップの作成	30
3.3.3 圧縮バックアップの作成	32
3.3.4 部分バックアップの作成	33
3.3.5 単一ファイルバックアップの作成	37
3.3.6 オプティミスティックバックアップの作成	40
3.3.7 インメモリーデータベースデータのバックアップの作成	41
3.3.8 スケジュールされたバックアップの作成	42
4 データベースのリカバリとリストア	43
4.1 バックアップをリストアする準備	43
4.2 リストア操作の実行	44
4.3 ホットバックアップからのポイントインタイム・リカバリ	46
4.4 単一の .ibd ファイルのバックアップとリストア	47
4.5 データベースのアップグレードまたはダウングレードによるバックアップのリストア	48
5 mysqlbackup コマンドリファレンス	51
5.1 mysqlbackup コマンド行オプション	52
5.1.1 サブコマンド	57
5.1.2 標準オプション	63
5.1.3 接続オプション	64
5.1.4 サーバリポジトリオプション	65
5.1.5 バックアップリポジトリオプション	67
5.1.6 メタデータオプション	70
5.1.7 圧縮オプション	70
5.1.8 増分バックアップオプション	71
5.1.9 部分バックアップとリストアオプション	74
5.1.10 単一ファイルバックアップオプション	80
5.1.11 パフォーマンス/スケーラビリティ/容量オプション	82
5.1.12 メッセージロギングオプション	88
5.1.13 進行状況レポートオプション	89
5.1.14 暗号化オプション	92
5.1.15 クラウドストレージオプション	93
5.1.16 特別なバックアップの種類オプション	94
5.2 構成ファイルとパラメータ	95
6 レプリケーションで MySQL Enterprise Backup を使用する	97
6.1 新しいレプリケーションスレーブのセットアップ	97
6.2 スレーブデータベースのバックアップとリストア	98
6.3 マスターデータベースのリストア	98
7 MySQL Enterprise Backup のパフォーマンスに関する考慮事項	101
7.1 バックアップパフォーマンスの最適化	101
7.2 リストアパフォーマンスの最適化	104
8 バックアップの暗号化	107
9 MySQL Enterprise Backup をメディア管理ソフトウェア (MMS) 製品と使用する	109
9.1 Oracle Secure Backup によるテープへのバックアップ	109
10 MySQL Enterprise Backup のトラブルシューティング	111
10.1 MySQL Enterprise Monitor によるバックアップのモニタリング	111
10.2 MySQL Enterprise Backup のエラーコード	111
10.3 破損の問題の回避	111

10.4 MySQL Enterprise Backup ログの使用	112
10.5 MySQL Enterprise Backup マニフェストの使用	114
11 MySQL Enterprise Backup のよくある質問	115

第 3 章 データベースサーバーのバックアップ

目次

3.1 最初のバックアップの前に	23
3.1.1 データベース情報を収集する	23
3.1.2 MySQL の権限をバックアップ管理者に与える	25
3.1.3 バックアップデータの場所を指定する	25
3.2 通常のバックアップ/検証/リストアのサイクル	26
3.2.1 MySQL インスタンス全体のバックアップ	26
3.2.2 バックアップの検証	28
3.2.3 元の場所でのデータベースのリストア	28
3.3 バックアップのシナリオと例	29
3.3.1 完全バックアップの作成	29
3.3.2 増分バックアップの作成	30
3.3.3 圧縮バックアップの作成	32
3.3.4 部分バックアップの作成	33
3.3.5 単一ファイルバックアップの作成	37
3.3.6 オプティミスティックバックアップの作成	40
3.3.7 インメモリーデータベースデータのバックアップの作成	41
3.3.8 スケジュールされたバックアップの作成	42

このセクションでは、MySQL Enterprise Backup で作成できるさまざまなバックアップの種類と、それらを生成するための手法について、`mysqlbackup` コマンドの関連する構文を示す例とともに説明します。`mysqlbackup` コマンドの完全な構文の参照も含まれます。

3.1 最初のバックアップの前に

バックアップのベストプラクティスには、計画と戦略が含まれます。このセクションでは、こうした計画と戦略を用意するために必要ないくつかの準備について概説します。

3.1.1 データベース情報を収集する

特定のデータベースサーバーをはじめてバックアップする前に、次の表に説明するように情報を集め、いくつかのディレクトリ名を決定します。

表 3.1 データベースのバックアップに必要な情報

収集する情報	確認する場所	使用方法
MySQL 構成ファイルへのパス	デフォルトのシステムの場合、ハードコードされたアプリケーションのデフォルトの場所、または <code>mysqld</code> 起動スクリプトの <code>--defaults-file</code> オプションから。	これは、 <code>--defaults-file</code> オプションを使用して、 <code>mysqlbackup</code> コマンドにデータベース構成情報を伝達する推奨の方法です。接続およびデータレイアウト情報が構成ファイルから得られる場合、次に挙げるその他のほとんどの選択肢を省略できます。
MySQL ポート	MySQL 構成ファイルまたは <code>mysqld</code> 起動スクリプト。	バックアップ操作中にデータベースインスタンスに接続するために使用されます。 <code>mysqlbackup</code> の <code>--port</code> オプションを介して指定されます。MySQL 構成ファイルから得られる場合、 <code>--port</code> は必要ありません。オフライン (コールド) バックアップを実行するときには必要ありません。このバックアップでは、OS レベルの

収集する情報	確認する場所	使用方法
		ファイルアクセス許可を使用して直接ファイルを操作します。
MySQL データディレクトリへのパス	MySQL 構成ファイルまたは <code>mysqld</code> 起動スクリプト。	バックアップ操作中にデータベースインスタンスからファイルを取得する場合や、リストア操作中にファイルをコピーしてデータベースインスタンスに戻す場合に使用されます。ホットバックアップとウォームバックアップの場合は、データベース接続から自動的に取得されます。コールドバックアップの場合は、MySQL 構成ファイルから取得されます。
権限を持つ MySQL ユーザーの ID とパスワード	自身のデータベースのインストール中にこれを記録するか、所有していないデータベースをバックアップしている場合は、DBA から取得します。オフライン (コールド) バックアップを実行するときには必要ありません。このバックアップでは、OS レベルのファイルアクセス許可を使用して直接ファイルを操作します。コールドバックアップの場合、管理ユーザーとしてログインします。	<code>mysqlbackup</code> の <code>--password</code> オプションを介して指定されます。パスワード引数なしに <code>--password</code> オプションが指定されている場合、端末から要求されます。
バックアップデータを格納するパス	これを選択します。詳細は、 セクション3.1.3「バックアップデータの場所を指定する」 を参照してください。	<code>mysqlbackup</code> がデータを書き込んでも、古いバックアップを上書きしたり別のバックアップからのデータと混在したりしないようにするため、デフォルトでこのディレクトリは空である必要があります。同じメインディレクトリ下に複数セットのバックアップデータを格納する場合、 <code>--with-timestamp</code> オプションを使用して、一意の名前でサブディレクトリを自動的に作成します。
バックアップファイルの所有者およびアクセス許可情報 (Linux、Unix、および OS X システムの場合)	MySQL データディレクトリ内。	別の OS ユーザー ID か、元のファイルに適用される別の <code>umask</code> 設定を使用してバックアップを行う場合、バックアップデータに対して <code>chown</code> や <code>chmod</code> などのコマンドを実行する必要がある可能性があります。詳細は、 セクションA.1「MySQL Enterprise Backup の制限」 を参照してください。
InnoDB Redo ログファイルのサイズ	<code>innodb_log_file_size</code> 構成変数と <code>innodb_log_files_in_group</code> 構成変数の値から計算されます。 <code>--incremental-with-redo-log-only</code> オプションについて説明した手法を使用します。	<code>--incremental</code> オプションではなく、 <code>--incremental-with-redo-log-only</code> オプションを使用して増分バックアップを実行する場合にのみ必要です。InnoDB Redo ログのサイズと Redo データの生成速度によって、増分バックアップを実行する必要がある頻度が決まります。
Redo データが生成される速度	さまざまな時点の InnoDB 論理シーケンス番号 の値から計算されます。 <code>--incremental-with-redo-log-only</code> オプションについて説明した手法を使用します。	<code>--incremental</code> オプションではなく、 <code>--incremental-with-redo-log-only</code> オプションを使用して増分バックアップを実行する場合にのみ必要です。InnoDB Redo ログのサイズと

収集する情報	確認する場所	使用方法
		Redo データの生成速度によって、増分バックアップを実行する必要がある頻度が決まります。

3.1.2 MySQL の権限をバックアップ管理者に与える

ほとんどのバックアップ操作では、`mysqlbackup` コマンドは、`--user` オプションと `--password` オプションを通じて MySQL Server に接続します。このユーザーは特定の権限を必要とします。最低限の権限セットを持つ新しいユーザーを作成するか、ルートユーザーなどの管理アカウントを使用することができます。

`mysqlbackup` が接続する MySQL ユーザーの最低限の権限は次のとおりです。

- すべてのデータベースおよびテーブルに対する `RELOAD`。
- `mysql.backup_progress` テーブルと `mysql.backup_history` テーブルに対する `CREATE`、`INSERT`、`DROP`、および `UPDATE` と、さらに `mysql.backup_history` に対する `SELECT`。
- ロギングを有効および無効にする場合と、データベース処理の中断を最小限に抑えるためにロックを最適化する場合の `SUPER`。
- バックアップで格納される `binlog` の位置を取得する場合の `REPLICATION CLIENT`。

ローカルホストから接続している MySQL ユーザー (この例では `mysqlbackup`) に対してこれらの権限を設定するには、`mysql` クライアントプログラムから次のようなステートメントを発行します。

```
GRANT RELOAD ON *.* TO 'mysqlbackup'@'localhost';
GRANT CREATE, INSERT, DROP, UPDATE ON mysql.backup_progress TO 'mysqlbackup'@'localhost';
GRANT CREATE, INSERT, SELECT, DROP, UPDATE ON mysql.backup_history TO 'mysqlbackup'@'localhost';
GRANT REPLICATION CLIENT ON *.* TO 'mysqlbackup'@'localhost';
GRANT SUPER ON *.* TO 'mysqlbackup'@'localhost';
FLUSH PRIVILEGES;
```

`トランスポータブルテーブルスペース (TTS)` を使用して InnoDB テーブルをバックアップおよびリストアするには、次の追加権限が必要です。

- テーブルをバックアップするための `LOCK TABLES` および `SELECT`
- テーブルをリストアするための `CREATE` および `ALTER`

これらの権限を設定するには、`mysql` クライアントプログラムから次のようなステートメントを発行します。

```
GRANT LOCK TABLES, SELECT, CREATE, ALTER ON *.* TO 'mysqlbackup'@'localhost';
FLUSH PRIVILEGES;
```

3.1.3 バックアップデータの場所を指定する

すべてのバックアップ関連操作は、バックアップデータを保持している指定のディレクトリ下で、新しいファイルを作成するか、既存のファイルを参照します。ストレージ領域が十分なファイルシステム上で、このディレクトリを前もって選択します。(別のサーバーからリモートでマウントすることもできます。) `mysqlbackup` コマンドの多くの呼び出しに対して `--backup-dir` オプションでこのディレクトリへのパスを指定します。

自動的なジョブによる定期的なバックアップスケジュールを確立したあとは、メインのバックアップディレクトリ下にあるタイムスタンプが記されたサブディレクトリ内にそれぞれのバックアップを保持することをお勧めします。 `mysqlbackup` コマンドでこれらのサブディレクトリを自動的に作成させるには、`mysqlbackup` を実行することに、`--with-timestamp` オプションを指定します。

データベースをクローニングしてレプリケーションスレーブを設定する場合など、1回かぎりのバックアップ操作の場合は、毎回、新しいディレクトリを指定することも、`mysqlbackup` の `--force` オプションを指定して古いバックアップファイルを上書きすることもできます。

3.2 通常バックアップ/検証/リストアのサイクル

バックアップを作成および使用する基本ステップについて説明するために、次の例で、完全バックアップを行い、バックアップディレクトリのデータファイルを調べ、続いてバックアップをリストアして破損や損失データの問題を修正する方法を示します。

3.2.1 MySQL インスタンス全体のバックアップ

この例では、説明のために、コマンド行ですべての必要なオプションを指定します。バックアップ手順をテストおよび標準化したあとで、一部のオプションを MySQL 構成ファイルに移動できます。これらのオプションは、データベースの接続情報と、バックアップデータの格納場所を指定します。`mysqlbackup` は複数の種類のバックアップ、リストア、バック/アンバック操作を実行できるため、最後のオプション `backup` で操作の種類を指定します。

この例では、最後のオプションを `backup-and-apply-log` として指定します。このオプションは、最初のバックアップ後に追加ステージを実行して、バックアップ操作中に行われたすべての変更を使用してすべての InnoDB テーブルを最新の状態にします。その結果、バックアップは即座にリストアする準備が整います。巨大またはビジーなデータベースのバックアップの場合、データベースサーバー上のロードを最低限に抑えるために、これらのステージを分割できます。つまり、最初に `backup` オプションを付けて `mysqlbackup` を実行し、別のサーバーにバックアップを転送し、続いて `apply-log` オプションを付けて `mysqlbackup` を実行して最後の処理を実行します。

出力には、バックアップ操作で使用されるすべてのパラメータが、データベース接続を使用して自動的に取得される複数のパラメータを含めて反映されます。このバックアップジョブの一意の ID は、`mysqlbackup` によってインスタンス内部に作成される特別なテーブルに記録され、これにより、長時間実行するバックアップをモニタリングし、以前のバックアップの結果を表示することができます。最後の出力セクションには、バックアップデータの場所が繰り返され、作成されたばかりの**完全バックアップ**に対して**増分バックアップ**を次回に実行するときを使用できる `LSN` 値が表示されます。

```
$ ./mysqlbackup --user=root -p --backup-dir=/home/admin/backups backup-and-apply-log

MySQL Enterprise Backup version 3.11.1 Linux-2.6.18-274.el5-i686 [2014/09/04]
Copyright (c) 2003, 2014, Oracle and/or its affiliates. All Rights Reserved.

mysqlbackup: INFO: Starting with following command line ...
./mysqlbackup --user=root -p --backup-dir=/home/admin/backups
backup-and-apply-log

mysqlbackup: INFO:
Enter password:
mysqlbackup: INFO: MySQL server version is '5.6.17-log'.
mysqlbackup: INFO: Got some server configuration information from running server.

IMPORTANT: Please check that mysqlbackup run completes successfully.
At the end of a successful 'backup-and-apply-log' run mysqlbackup
prints "mysqlbackup completed OK!".

140904 12:34:54 mysqlbackup: INFO: MEB logfile created at /home/admin/backups/meta/MEB_2014-09-04.12-34-54_backup_apply_log.log

-----
Server Repository Options:
-----
datadir = /var/lib/mysql/
innodb_data_home_dir =
innodb_data_file_path = ibdata1:12M:autoextend
innodb_log_group_home_dir = /var/lib/mysql/
innodb_log_files_in_group = 2
innodb_log_file_size = 50331648
innodb_page_size = 16384
innodb_checksum_algorithm = innodb
innodb_undo_directory = /var/lib/mysql/
innodb_undo_tablespaces = 0
innodb_undo_logs = 128

-----
Backup Config Options:
-----
datadir = /home/admin/backups/datadir
innodb_data_home_dir = /home/admin/backups/datadir
```

```
innodb_data_file_path = ibdata1:12M:autoextend
innodb_log_group_home_dir = /home/admin/backups/datadir
innodb_log_files_in_group = 2
innodb_log_file_size = 50331648
innodb_page_size = 16384
innodb_checksum_algorithm = innodb
innodb_undo_directory = /home/admin/backups/datadir
innodb_undo_tablespaces = 0
innodb_undo_logs = 128

mysqlbackup: INFO: Unique generated backup id for this is 14098484948398421

mysqlbackup: INFO: Creating 14 buffers each of size 16777216.
140904 12:34:57 mysqlbackup: INFO: Full Backup operation starts with following threads
    1 read-threads  6 process-threads  1 write-threads
140904 12:34:57 mysqlbackup: INFO: System tablespace file format is Antelope.
140904 12:34:57 mysqlbackup: INFO: Starting to copy all innodb files...
140904 12:34:57 mysqlbackup: INFO: Found checkpoint at lsn 1611237.
140904 12:34:57 mysqlbackup: INFO: Starting log scan from lsn 1610752.
140904 12:34:57 mysqlbackup: INFO: Copying log...
140904 12:34:57 mysqlbackup: INFO: Log copied, lsn 1611237.
140904 12:34:57 mysqlbackup: INFO: Copying /var/lib/mysql/ibdata1 (Antelope file format).
140904 12:34:59 mysqlbackup: INFO: Completing the copy of innodb files.
140904 12:34:59 mysqlbackup: INFO: Starting to copy Binlog files...
140904 12:34:59 mysqlbackup: INFO: Preparing to lock tables: Connected to mysqld server.
140904 12:34:59 mysqlbackup: INFO: Starting to lock all the tables...
140904 12:35:00 mysqlbackup: INFO: All tables are locked and flushed to disk
140904 12:35:00 mysqlbackup: INFO: Opening backup source directory '/var/lib/mysql/'
140904 12:35:00 mysqlbackup: INFO: Starting to backup all non-innodb files in
    subdirectories of '/var/lib/mysql/'
140904 12:35:00 mysqlbackup: INFO: Copying the database directory 'performance_schema'
140904 12:35:01 mysqlbackup: INFO: Completing the copy of all non-innodb files.
140904 12:35:01 mysqlbackup: INFO: Completed the copy of binlog files...
140904 12:35:02 mysqlbackup: INFO: A copied database page was modified at 1611237.
    (This is the highest lsn found on page)
    Scanned log up to lsn 1611237.
    Was able to parse the log up to lsn 1611237.
    Maximum page number for a log record 0
140904 12:35:02 mysqlbackup: INFO: All tables unlocked
140904 12:35:02 mysqlbackup: INFO: All MySQL tables were locked for 2.530 seconds.
140904 12:35:03 mysqlbackup: INFO: Reading all global variables from the server.
140904 12:35:03 mysqlbackup: INFO: Completed reading of all global variables from the server.
140904 12:35:03 mysqlbackup: INFO: Creating server config files server-my.cnf and server-all.cnf in /home/admin/backups
140904 12:35:03 mysqlbackup: INFO: Full Backup operation completed successfully.
140904 12:35:03 mysqlbackup: INFO: Backup created in directory '/home/admin/backups'
140904 12:35:03 mysqlbackup: INFO: MySQL binlog position: filename mysqld-bin.000001, position 120
```

Parameters Summary

Start LSN	: 1610752
End LSN	: 1611237

```
mysqlbackup: INFO: Creating 14 buffers each of size 65536.
140904 12:35:03 mysqlbackup: INFO: Apply-log operation starts with following threads
    1 read-threads  1 process-threads
mysqlbackup: INFO: Using up to 100 MB of memory.
140904 12:35:03 mysqlbackup: INFO: ibbackup_logfile's creation parameters:
    start lsn 1610752, end lsn 1611237,
    start checkpoint 1611237.
mysqlbackup: INFO: InnoDB: Starting an apply batch of log records to the database...
InnoDB: Progress in percent: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
mysqlbackup: INFO: InnoDB: Setting log file size to 50331648
mysqlbackup: INFO: InnoDB: Setting log file size to 50331648
140904 12:35:14 mysqlbackup: INFO: We were able to parse ibbackup_logfile up to
    lsn 1611237.
mysqlbackup: INFO: Last MySQL binlog file position 0 120, file name mysqld-bin.000001:120
140904 12:35:14 mysqlbackup: INFO: The first data file is '/home/admin/backups/datadir/ibdata1'
    and the new created log files are at '/home/admin/backups/datadir'
140904 12:35:14 mysqlbackup: INFO: Apply-log operation completed successfully.
140904 12:35:14 mysqlbackup: INFO: Full backup prepared for recovery successfully.

mysqlbackup completed OK!
```

これで、バックアップサブディレクトリは、指定した `backup-dir` の下に作成されます。新しい各バックアップのディレクトリ名は、バックアップの実行が開始した時点の、地域のタイムゾーンでの日付とクロック時間から形成されます。バックアップディレクトリには、バックアップされた `ibdata` ファイルと `ibbackup_logfile` ファイルが含まれます。それぞれのサブディレクトリは MySQL データベースに対応し、`.frm`、`.MYD`、`.MYI`、および同様のファイルのコピーを含みます。このようなバックアップディレクトリのレイアウトの例については、[セクションC.1「完全バックアップのディレクトリ構造例」](#)を参照してください。

3.2.2 バックアップの検証

バックアップを検証するには、別のサーバーでバックアップデータをリストアし、新しいデータディレクトリで MySQL デーモン (`mysqld`) を実行します。続いて、`SHOW` ステートメントを実行し、データベースおよびテーブル構造を検証し、クエリーを実行して行数や最新の更新などを検証できます。

これは、バックアップデータをほかの用途で利用したい場合に使用するのと同じ、一般的な手法です。たとえば、マスターサーバーのバックアップを作成することでレプリケーションスレーブを設定したり、バックアップを新しい MySQL インスタンスに変換してレポートクエリーを実行したりすることが考えられます。

注記

必ず、`datadir` でバックアップディレクトリを指し示して `mysqld` を実行するのではなく、リストアされたデータに対して検証を行なってください。データを検証するために使用する SQL ステートメントは、ベースとなる論理シーケンス番号を変更します。これにより、以降の増分バックアップにバックアップディレクトリを使用できなくなります。

前の例のように `backup-and-apply-log` オプションを使用してバックアップを行なった場合、バックアップデータは完全に整合し、検証する準備ができています。`backup` オプションを使用して最初のステージだけを実行した場合、この検証を行う前に、`apply-log` オプションを使用して `mysqlbackup` を2度目に実行します。(通常、この2番目のフェーズは、ほかのサーバーでバックアップデータを転送したあとに実行して、元のデータベースサーバーでのロードを最低限に抑えます。)

別のサーバーでデータベースファイルをリストアする手順については、[第4章「データベースのリカバリとリストア」](#)を参照してください。

リストアしたデータに対して `mysqld` デーモンを実行するには、有効な構成ファイルが必要です。これは、`mysqld` コマンドの `--defaults-file` オプションで指定します。元の `my.cnf` ファイルからのほとんどの設定は、`mysqlbackup` で必要なパラメータの小さなサブセットだけを含む、バックアップディレクトリ内の `backup-my.cnf` ファイルと組み合わせて再利用できます。これらの2つのファイルを新しいファイルに連結することによって、新しい構成ファイルを作成し、検証を行うサーバーでその構成ファイルを使用します。`datadir` パラメータが検証サーバー上の正しい場所を確実に指し示すように、結果として得られたファイルを編集します。検証サーバーで別の接続設定を使用する必要がある場合は、ポートやソケットなどの値を編集します。

3.2.3 元の場所でのデータベースのリストア

バックアップから MySQL インスタンスをリストアするには:

- `mysqladmin shutdown` コマンドなど、通常的手法を使用して、データベースサーバーをシャットダウンします。
- `backup-and-apply-log` オプションを使用してバックアップを実行するか、最初のバックアップのあとに `apply-log` オプションを付けて `mysqlbackup` を実行して、バックアップデータが完全に整合していることを確認します。
- `copy-back` オプションを付けて `mysqlbackup` コマンドを使用します。この操作は、テーブル、インデックス、メタデータ、およびほかの必要なファイルをコピーして、元の MySQL 構成ファイルの定義に従って元の場所に戻します。この操作の一部として指定できるオプションのさまざまな組み合わせについては、[セクション5.1.1.3「既存のバックアップのリストア」](#)を参照してください。

```
$ mysqlbackup --defaults-file=path_to_my.cnf \
--datadir=path_to_data_directory \
--backup-dir=path_to_backup_directory copy-back
...many lines of output...
mysqlbackup: Finished copying backup files.
```



```
101208 16:48:13 mysqlbackup: mysqlbackup completed OK!
```

これで、元のデータベースディレクトリがバックアップからリストアされ、データベースサーバーを再起動できます。

3.3 バックアップのシナリオと例

次のタスクと例はすべて、`mysqlbackup` コマンドを使用します。構文の詳細は、[第5章「mysqlbackup コマンドリファレンス」](#)を参照してください。

3.3.1 完全バックアップの作成

ほとんどのバックアップ戦略では、MySQL Server の完全なバックアップから開始し、そこからすべてのデータベースとテーブルをリストアできます。**完全バックアップ**を一度行なったあと、次の複数のバックアップジョブでは**増分バックアップ** (より小さく高速) を行うことができます。その後は、定期的に完全バックアップを新しく行なって、サイクルを再度開始します。

このセクションでは、この非常に基本的なバックアップに関するいくつかの考慮事項について説明します。完全バックアップはほかの種類のバックアップよりも時間がかかり、大きなバックアップファイルが作成される可能性があるため、速度、容量、および利便性に関する判断が、バックアップ戦略のこの部分にとって特に重要になります。

完全バックアップを作成するコマンドを示した例については、[セクション3.2.1「MySQL インスタンス全体のバックアップ」](#)を参照してください。

コマンド行のオプションか、構成ファイルのオプションか？

明確にするために、このマニュアルの例では通常、コマンド行のオプションを示すことで、それぞれのバックアップジョブで同じである可能性のある接続パラメータとほかの情報について説明します。便宜上、また、一貫性を保つために、`mysqlbackup` コマンドに渡す MySQL 構成ファイルの「`mysqlbackup`」セクションに、これらのオプションを含めることができます。`mysqlbackup` はまた、これらのオプションが存在する場合は、「`mysqld`」セクションから選択することもできます。たとえば、構成ファイル内のポート情報に依存すると、データベースインスタンスが別のポートに切り替えた場合に、バックアップスクリプトを編集する必要はなくなります。

単一のディレクトリでの出力か、タイムスタンプが記されたサブディレクトリでの出力か？

便宜上、`--with-timestamp` オプションは、各バックアップジョブからの出力を保持するために、バックアップディレクトリ下に一意の名前を付けたサブディレクトリを作成します。タイムスタンプが記されたサブディレクトリであれば、たとえば一定の有効期間を過ぎたバックアップデータを削除またはアーカイブすることによって、保持期間の確立がより簡単になります。このオプションはデフォルトでは設定されていません。

単一のバックアップディレクトリを使用する場合 (つまり、`--with-timestamp` オプションを省略した場合)、バックアップジョブごとに新しい一意のディレクトリ名を指定するか、`--force` オプションを指定して既存のバックアップファイルを上書きしてください。

各増分バックアップコマンドの一部に `--incremental-base` オプションを使用して、以前のバックアップを含むディレクトリを指定します。ディレクトリ名を予測できる名前にするには、`--with-timestamp` オプションを省略して、代わりにバックアップスクリプトの一部としてディレクトリ名のシーケンスを生成します。

常に完全バックアップか、完全バックアップに増分バックアップを組み合わせるか？

InnoDB データボリュームが小さい場合、またはデータベースが非常にビジーであるために高い割合のデータがバックアップ間で変更した場合、毎回、完全バックアップを実行できます。通常、定期的に完全バックアップを実行し、[セクション3.3.2「増分バックアップの作成」](#)の説明に従って2つの完全バックアップの合間に増分バックアップを実行することによって、時間とストレージ領域を節約できます。

圧縮を使用するか、使用しないか？

圧縮バックアップを作成すると、ストレージ領域を大幅に節約し、I/O を大幅に軽減できます。また、LZ4 圧縮方法 (リリース 3.10 以降で導入) を使用すると、圧縮処理のオーバーヘッドは非常に低くなります。データベースバックアップが、アクティブなデータベースファイルが置かれているより高速なディスクシステムから、低速である可能性があるストレージに移動している場合は、圧縮によって全体のバックアップ時間が大幅に短縮されることがよく

あります。リストア時間が短縮する結果になることもあります。LZ4 ベースのバックアップはより短時間で終わる場合が多いため、通常、ほとんどのユーザーには圧縮なしよりも LZ4 圧縮をお勧めします。ただし、環境内で MySQL Enterprise Backup をテストして、もっとも効率的なアプローチを判断してください。

3.3.2 増分バックアップの作成

増分バックアップは、前回のバックアップ以降に変更のあったデータだけをバックアップします。この手法は、バックアップ戦略の設計での柔軟性を高め、バックアップに必要なストレージを軽減します。

増分バックアップは通常、完全バックアップより小さく、時間がかからないため、頻繁なバックアップジョブに適しています。増分バックアップを頻繁に取得しておけば、常にデータベースを、数時間または数日前と同じ状態にリストアでき、完全バックアップを頻繁に取得する場合と同じだけのデータベースサーバーでのロードまたはストレージオーバーヘッドは不要になります。

注記

増分バックアップは常に既存のバックアップファイルのセットに追加するため、増分バックアップを行う前に、少なくとも 1 つの**完全バックアップ**を作成してください。

増分バックアップは、`mysqlbackup` コマンドのオプションで有効になります。単純な増分バックアップの場合、`--incremental` オプションを指定します。代替の方法では、`--incremental-with-redo-log-only` オプションを使用しますが、ユーザー側で追加の計画が必要になります。

また、前の完全または増分バックアップの特定の時点を指定します。便宜上、`--incremental-base` オプションを使用して、前のバックアップディレクトリに格納されたメタデータから必要な**ログシーケンス番号** (LSN) を自動的に取得できます。または、`--start-lsn` オプションを使用し、前の完全または増分バックアップから終了 LSN を使用して明確な LSN 値を指定できます。

バックアップデータをリストアする準備を行うには、各増分バックアップを元の完全バックアップと組み合わせます。通常、指定された期間の経過後に新しい完全バックアップを実行し、そのあとで古い増分バックアップデータを破棄できます。

増分バックアップの「ログの適用」手順を実行するときに、オプションシーケンス `--incremental apply-log` と、2 つの MySQL 構成ファイルへのパス (最初に、更新している完全バックアップを指す `.cnf` ファイルと、次に増分バックアップデータファイルを指す `.cnf` ファイル) を指定します。前回の完全バックアップ以降に複数の増分バックアップを取得していた場合は、このような「ログの適用」ステップを順に複数実行して、完全バックアップを完全に最新の状態にすることができます。

増分バックアップの領域に関する考慮事項

増分バックアップ機能は主に、InnoDB テーブルや、読み取り専用またはまれにしか更新されない非 InnoDB テーブルを対象としています。非 InnoDB ファイルでは、前回のバックアップ以降にファイルに変更があった場合、ファイル全体が増分バックアップに含まれます。

`--compress` オプションでは増分バックアップを実行できません。

増分バックアップは、テーブルの行ではなく、InnoDB **データファイル**内の**ページ**のレベルで変更を検出します。変更のあった各ページがバックアップされます。したがって、領域と時間の節約は、変更された InnoDB の行またはカラムの割合に正確には比例しません。

増分バックアップの例

この例では、`mysqlbackup` コマンドを使用して、すべてのデータベースとテーブルを含む MySQL Server の増分バックアップを作成します。`--incremental-base` オプションを使用したものと、`--start-lsn` オプションを使用したものの 2 つの代替例を示します。

`--incremental-base` オプションを使用した場合、あるバックアップと次のバックアップの間で LSN 値を追跡する必要はありません。代わりに、前回のバックアップ (完全または増分) のディレクトリを指定し、`mysqlbackup` が、以前のバックアップのメタデータに基づいてこのバックアップの開始点を算出します。既知のディレクトリ名のセットが必要なため、`--with-timestamp` オプションを使用するのではなく、ハードコード化された名前を使用するか、自身のバックアップスクリプトで名前のシーケンスを生成することができます。


```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --incremental \
--incremental-base=dir:/incr-backup/wednesday \
--incremental-backup-dir=/incr-backup/thursday \
backup
...many lines of output...
mysqlbackup: Backup created in directory '/incr-backup/thursday'
mysqlbackup: start_lsn: 2654255717
mysqlbackup: incremental_base_lsn: 2666733462
mysqlbackup: end_lsn: 2666736714

101208 17:14:58 mysqlbackup: mysqlbackup completed OK!
```

`--start-lsn` オプションでは、前回のバックアップの LSN を記録する必要がありますが、前回のバックアップの場所があまり重要でないため、`--with-timestamp` を使用して、名前の付けられたサブディレクトリを自動的に作成することができます。

```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --incremental \
--start-lsn=2654255716 \
--with-timestamp \
--incremental-backup-dir=/incr-backup \
backup
...many lines of output...
mysqlbackup: Backup created in directory '/incr-backup/2010-12-08_17-14-48'
mysqlbackup: start_lsn: 2654255717
mysqlbackup: incremental_base_lsn: 2666733462
mysqlbackup: end_lsn: 2666736714

101208 17:14:58 mysqlbackup: mysqlbackup completed OK!
```

`--incremental` オプションを使用するなどの箇所でも、代わりに `--incremental-with-redo-log-only` オプションを使用できます。`--incremental-with-redo-log-only` は、`--incremental` オプションより正確な LSN に依存する程度が大きいため、この種の増分バックアップには、`--start-lsn` オプションではなく `--incremental-base` オプションを使用してください。

このような代替りの増分バックアップが機能するには、前回の増分バックアップ以降のすべての変更が Redo ログに存在し、上書きされないようにする必要があります。変更された情報が十分に少量で、Redo ログファイルが十分に大きいことが必要です。これらの要件を検証する方法については、`--incremental-with-redo-log-only` オプションの説明を参照してください。

```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --incremental \
--incremental-base=dir:/incr-backup/wednesday \
--incremental-backup-dir=/incr-backup/thursday \
backup
...many lines of output...
mysqlbackup: Backup created in directory '/incr-backup/thursday'
mysqlbackup: start_lsn: 2654255717
mysqlbackup: incremental_base_lsn: 2666733462
mysqlbackup: end_lsn: 2666736714

101208 17:14:58 mysqlbackup: mysqlbackup completed OK!
```

通常の増分バックアップからのファイルの一覧については、[セクションC.3「増分バックアップのディレクトリ構造例」](#)を参照してください。

もう一度、バックアップを実行していた間に行われた変更を、完全バックアップに適用します。

```
$ mysqlbackup --backup-dir=/full-backup/2010-12-08_17-14-11 apply-log
..many lines of output...
101208 17:15:10 mysqlbackup: Full backup prepared for recovery successfully!

101208 17:15:10 mysqlbackup: mysqlbackup completed OK!
```

続いて、増分バックアップから変更を適用します。

```
$ mysqlbackup --incremental-backup-dir=/incr-backup/2010-12-08_17-14-48
--backup-dir=/full-backup/2010-12-08_17-14-11 apply-incremental-backup
...many lines of output...
101208 17:15:12 mysqlbackup: mysqlbackup completed OK!
```

これで、完全バックアップディレクトリ内のデータファイルは全面的に、前回の増分バックアップ時点の最新の状態になっています。

この例には増分バックアップが示されています。前回実行した完全バックアップで、もっとも高い LSN が 2638548215 だったことがレポートされました。

```
mysqlbackup: Was able to parse the log up to lsn 2638548215
```

ここではコマンドでもう一度その番号を指定します。増分バックアップには、指定した LSN の後に行われたすべての変更が含まれます。

```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --incremental \
--start-lsn=2638548215 \
--incremental-backup-dir=/incr-backup/2010-12-08_17-14-48 \
--backup-dir=/full-backup/2010-12-08_17-14-11 \
backup
...many lines of output...
mysqlbackup: Scanned log up to lsn 2654252454.
mysqlbackup: Was able to parse the log up to lsn 2654252454.
mysqlbackup: Maximum page number for a log record 0
mysqlbackup: Backup contains changes from lsn 2638548216 to lsn 2654252454
101208 17:12:24 mysqlbackup: Incremental backup completed!
```

次のステップ:

- `mysqlbackup: Was able to parse the log up to lsn LSN_number` などの、バックアップの終わりに表示されるメッセージ内の LSN 値を書き留めてください。この増分バックアップのあとに行われた変更の増分バックアップを実行するときに、この値を指定します。
- いつでもバックアップをリストアできるように、バックアップファイルに増分バックアップを適用します。最初にバックアップデータを別のサーバーに移動して、データベースサーバー自体でのこの操作の CPU および I/O オーバーヘッドを回避できます。
- 定期的に、データベースアクティビティーの日付または量で判断し、さらに増分バックアップを取得します。
- オプションで、完全、非圧縮、または圧縮バックアップを取得することによって、サイクルを定期的に開始しなおしてください。通常、このマイルストーンは、もっとも古いバックアップデータをアーカイブおよびクリアできるに行われます。

3.3.3 圧縮バックアップの作成

ディスク領域を節約するために、`mysqlbackup` の `--compress` オプションを使用して、InnoDB バックアップデータファイルを圧縮できます。圧縮を使用すると、より多くのバックアップデータを保有でき、別のサーバーへのバックアップデータの転送時間を短縮できます。マイナス面としては、バックアップ自体を行う間の CPU オーバーヘッドの増大と、データを圧縮解除するためのリストアプロセスに必要な時間の増加があります。

バックアップ圧縮機能は、InnoDB テーブルでのみ機能します。バックアップ中に InnoDB テーブルスペースファイルを圧縮したあと、これらのファイルには、通常の `.ibd` 拡張子でなく `.ibz` 拡張子が付けられます。追加のディスク領域を節約せずに CPU サイクルを浪費することを避けるために、`--compress` は、Barracuda ファイル形式を使用した圧縮済みのテーブルを圧縮しようとしません。このようなテーブルスペースファイルは通常の `.ibd` 拡張子のままです。

注記

InnoDB テーブルスペース内に未使用領域がある場合、非圧縮バックアップ中にファイル全体がコピーされます。未使用領域のストレージオーバーヘッドを避けるため、圧縮バックアップを実行してください。

`--compress` オプションは、完全バックアップだけに使用でき、増分バックアップには使用できません。

`--compress-method` オプションで、使用する圧縮アルゴリズムを選択することもできます。ZLIB または LZMA 圧縮アルゴリズムを使用する場合は、`--compress-level` オプションで圧縮のレベルも選択できます。詳細は、[セクション 5.1.7 「圧縮オプション」](#)を参照してください。

圧縮バックアップを作成するためのコマンド例は次のとおりです。

```
mysqlbackup --defaults-file=/etc/my.cnf --compress --compress-level=5 backup
```

単一ファイルの圧縮バックアップを作成するためのコマンド例は次のとおりです。

```
mysqlbackup --defaults-file=/etc/my.cnf --compress --compress-level=5 \
--backup-image=backup.img backup-to-image
```

次のステップ:

- 完全バックアップと増分バックアップの終わりに表示されるメッセージ内の (たとえば `mysqlbackup: Was able to parse the log up to lsn LSN_number` の行内にある) LSN 値を書き留めてください。この完全バックアップのあとに行われた変更の増分バックアップを実行するときに、この値を指定します。
- いつでも完全バックアップをリストアできるよう、圧縮バックアップファイルに**ログを適用**します。最初にバックアップデータを別のサーバーに移動して、データベースサーバーでこの操作を実行する CPU および I/O オーバーヘッドを回避できます。
- ログを適用したあと、定期的に**増分バックアップを取得**します。これは完全バックアップより小さく、すばやく作成できます。

3.3.4 部分バックアップの作成

注記

部分バックアップの作成を容易にするため、MySQL Enterprise Backup 3.10 では、`--include-tables` と `--exclude-tables` という部分バックアップ用の 2 つの新しいオプションが導入されています。これらの新しいオプションは、`--include`、`--databases`、`--databases-list-file`、および `--only-innodb-with-frm` の古いオプションに置き換わるものです。これらの古いオプションは新しいオプションと互換性がなく、次回のリリースで非推奨になります。次の説明では、この新しいオプションが部分バックアップに使用されることを想定しています。参考のため、古いオプションに関する情報をこのセクションの最後の「**レガシーオプションによる部分バックアップの作成**」に記載しています。

デフォルトではデータディレクトリ内のすべてのファイルがバックアップに含まれるため、このディレクトリ内のすべての MySQL ストレージエンジン、サードパーティ製ストレージエンジン、さらにはデータベース以外のファイルからのデータも、バックアップに含まれます。このセクションでは、データをバックアップするか除外するかを選択する場合に使用できるオプションについて説明します。

MySQL Enterprise Backup には、次のようなさまざまな部分バックアップを作成する複数の方法が用意されています。

- 特定のテーブルをその名前で含めるまたは除外する。これは、`--include-tables` オプションまたは `--exclude-tables` オプションを使用します。
`--include-tables` オプションまたは `--exclude-tables` オプションで指定された正規表現と照合して各テーブルがチェックされます。正規表現がテーブルの完全修飾名に一致すると (`db_name.table_name` の形式)、そのテーブルがバックアップに含まれるか除外されます。使用される正規表現構文は、POSIX 1003.2 規格で指定された拡張形式です。これらのオプションは、Henry Spencer の正規表現ライブラリとともに実装されています。
- InnoDB テーブルの一部またはすべてを含めるが、ほかのテーブルタイプは含めない。これは、`--only-innodb` オプションを使用します。
- MySQL データディレクトリに存在するが、実際には MySQL インスタンスの一部ではないファイルを除外する。これは、`--only-known-file-types` オプションを使用します。
- 前述のオプションの組み合わせを使用して、複数の選択結果を得る。
- **トランスポートブルテーブルスペース (TTS)** を使用して選択した InnoDB テーブルをバックアップする。これは、`--use-tts` オプションと `--include-tables` オプションまたは `--exclude-tables` オプション (あるいは両方) を使用します。

関連するすべてのオプションの構文の詳細は、**セクション 5.1.9 「部分バックアップとリストアオプション」** を参照してください。

重要

通常、部分バックアップは、完全な MySQL インスタンスを構築するために必要な相互関連要素がバックアップデータに含まれていない場合があるため、完全バックアップよりもリス

トアが困難です。特に、InnoDB テーブルが保持する内部 ID とその他のデータ値は、同じインスタンスにのみリストアでき、別の MySQL Server ではリストアできません。必ず、関連した手順と制限について理解するために、どの部分バックアップでもリカバリ手順を十分にテストしてください。

重要

InnoDB システムテーブルスペースは、インスタンス内のすべてのデータベースからの InnoDB テーブルに関するメタデータを保持しているため、ほかのデータベースを含むサーバー上で部分バックアップをリストアすると、システムがほかのデータベース内のこれらの InnoDB テーブルを追跡できなくなる可能性があります。部分バックアップは必ず、維持する必要のあるほかの InnoDB テーブルを含まない、新規の MySQL Server インスタンスでリストアしてください。

次に、部分バックアップのコマンド例を示します。

「emp」で始まる名前のテーブルをすべてバックアップに含める場合:

```
$ mysqlbackup \
--host=localhost --user=mysqluser --protocol=TCP --port=3306 \
--backup-dir=$MEB_BACKUPS_DIR/backupdir \ --include-tables='.emp' \
backup
```

「mysql」データベースと「performance_schema」データベースからのテーブルを除き、すべてのテーブルのバックアップを取得する場合:

```
$ mysqlbackup \
--host=localhost --user=mysqluser --protocol=TCP --port=3306 \
--backup-dir=$MEB_BACKUPS_DIR/backupdir \
--exclude-tables='^(mysql|performance_schema)\.' \
backup
```

「sales」データベース内のすべてのテーブルのバックアップを取得するが、「hardware」という名前のテーブルを除く場合:

```
$ mysqlbackup \
--host=localhost --user=mysqluser --protocol=TCP --port=3306 \
--backup-dir=$MEB_BACKUPS_DIR/backupdir \
--include-tables='^sales\.' --exclude-tables='^sales\.hardware$' \
backup
```

すべての InnoDB テーブルをバックアップするが、.frm ファイルはバックアップしない場合:

```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --only-innodb backup
```

適切なオプションを追加することにより、**圧縮**、**単一イメージ**、およびほかの種類の選択的バックアップも作成できます。

次のステップ:

- 完全バックアップと増分バックアップの終わりに表示される、[mysqlbackup: Was able to parse the log up to lsn LSN_number](#) などのメッセージ内の LSN 値を書き留めてください。この完全バックアップのあとに行われた変更の増分バックアップを実行するときに、この値を指定します。
- いつでも完全バックアップをリストアできるよう、非圧縮バックアップファイルに**ログを適用**します。最初にバックアップデータを別のサーバーに移動して、データベースサーバーでこの操作を実行する CPU および I/O オーバーヘッドを回避できます。
- ログを適用したあと、定期的に**増分バックアップ**を取得します。これらのような完全バックアップより非常に高速で、小さくなります。

レガシーオプションによる部分バックアップの作成

重要

このサブセクションの情報は、`--include`、`--databases`、`--databases-list-file`、および `--only-innodb-with-frm` のレガシーオプションを使用するためだけのもので、これらのオプションは

次回の発行時に非推奨になります。部分バックアップの作成には、代わりに `--include-tables` と `--exclude-tables` の新しいオプションを使用することを強くお勧めします。部分バックアップのレガシーオプションと新しいオプションを単一のコマンドで組み合わせることはできません。

MySQL Enterprise Backup では、部分バックアップのレガシーオプションを使用して、次のようなさまざまな部分バックアップを作成できます。

- 特定の InnoDB テーブルを含めるが、ほかのテーブルは含めない。この操作には、`--include`、`--only-innodb`、および `--only-innodb-with-frm` のオプションが含まれます。
- 選択したデータベースからの特定の非 InnoDB テーブルを含めるが、それ以外は含めない。この操作には、`--databases` オプションと `--databases-list-file` オプションが含まれます。

これらのすべてのオプションに関する構文の詳細は、「[レガシー部分バックアップオプション](#)」を参照してください。

注記

通常、部分バックアップは、完全な MySQL インスタンスを構築するために必要な相互関連要素がバックアップデータに含まれていない場合があるため、完全バックアップよりもリストアが困難です。特に、InnoDB テーブルが保持する内部 ID とその他のデータ値は、同じインスタンスのみリストアでき、別の MySQL Server ではリストアできません。必ず、関連した手順と制限について理解するために、どの部分バックアップでもリカバリ手順を十分にテストしてください。

`mysqlbackup` は、その `--include` オプションを使用して、一部の InnoDB テーブルを含み、ほかのテーブルは含まないバックアップを作成できます。

- `--include` オプションによる部分バックアップには、常に、InnoDB システムテーブルスペースとその内部のすべてのテーブルが含まれます。
- システムテーブルスペース外に格納された InnoDB テーブルの場合、部分バックアップには、その名前が、`--include` オプションで指定された正規表現に一致するテーブルだけが含まれます。

この操作には、除外されるテーブルが、個別の `table_name.ibd` ファイルに格納されることが必要です。InnoDB テーブルをシステムテーブルスペース外に置くには、`innodb_file_per_table` MySQL 構成オプションが有効な間に InnoDB テーブルを作成します。それぞれの `.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だけが保持されます。

`innodb_file_per_table` をオフにして作成されたこれらの InnoDB テーブルは、InnoDB システムテーブルスペースに通常どおりに格納され、バックアップから除外できません。

per-table データファイルを含むテーブルごとに、`db_name.table_name` の形式の文字列が、`--include` オプションで指定された正規表現と照合してチェックされます。正規表現が文字列 `db_name.table_name` 全体に一致した場合、テーブルはバックアップに含まれます。使用される正規表現構文は、POSIX 1003.2 規格で指定された拡張形式です。Unix のようなシステムでは、適宜、正規表現を引用符で囲んで、シェルメタキャラクタが解釈されないようにしてください。この機能は、Henry Spencer の正規表現ライブラリとともに実装されています。

生成されたバックアップディレクトリには、バックアップログファイルと InnoDB データファイルのコピーが含まれます。

重要: `mysqlbackup` コマンドは部分バックアップの取得をサポートしますが、部分バックアップからデータベースをリストアするときには注意してください。`mysqlbackup` は、たとえば `--databases` オプションを使用して部分バックアップを実行したときを除き、バックアップに含まれないテーブルの `.frm` ファイルもコピーします。`--include` オプションを付けて `mysqlbackup` を使用する場合は、データベースのリストア前に、バックアップに含まれないすべてのテーブルの `.frm` ファイルをバックアップデータから削除してください。

重要: InnoDB システムテーブルスペースは、インスタンス内のすべてのデータベースからの InnoDB テーブルに関するメタデータを保持しているため、ほかのデータベースを含むサーバー上で部分バックアップをリストアすると、システムがほかのデータベース内のこれらの InnoDB テーブルを追跡できなくなる可能性があります。部分バックアップは必ず、維持する必要があるほかの InnoDB テーブルを含まない、新規の MySQL Server インスタンスでリストアしてください。

`--only-innodb` オプションと `--only-innodb-with-frm` オプションは、InnoDB テーブルだけをバックアップし、ほかのストレージエンジンのテーブルはスキップします。これらを `--include` オプションと一緒に使用して、ほかのストレージエンジンが作成したほかのすべてのファイルを除きながら、InnoDB テーブルの選択的バックアップを作成することもできます。

例 3.1 InnoDB テーブルの非圧縮部分バックアップの作成

この例では、一部の InnoDB テーブルが独自のテーブルスペースを持つように MySQL を構成しました。ib で始まる名前を持つ、`test` データベース内の InnoDB テーブルだけを含む部分バックアップを作成します。`test` データベースのデータベースディレクトリの内容を次に示します。このディレクトリには、データベースの各テーブル (`alex1`、`alex2`、`alex3`、`blobt3`、`ibstest0`、`ibstest09`、`ibtest11a`、`ibtest11b`、`ibtest11c`、および `ibtest11d`) の MySQL 説明ファイル (`.frm` ファイル) が含まれます。これら 10 個のテーブルのうち、6 つ (`alex1`、`alex2`、`alex3`、`blobt3`、`ibstest0`、`ibstest09`) は per-table データファイル (`.ibd` ファイル) に格納されます。

```
$ ls /sqldata/mts/test
alex1.frm alex2.ibd blobt3.frm  ibstest0.ibd ibtest11a.frm ibtest11d.frm
alex1.ibd alex3.frm blobt3.ibd  ibtest09.frm ibtest11b.frm
alex2.frm alex3.ibd ibstest0.frm ibtest09.ibd ibtest11c.frm
```

`--include` オプションを付けて `mysqlbackup` を実行します。

```
# Back up some InnoDB tables but not any .frm files.
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --include='test\ib.*' --only-innodb backup
...many lines of output...
mysqlbackup: Scanned log up to lsn 2666737471.
mysqlbackup: Was able to parse the log up to lsn 2666737471.
mysqlbackup: Maximum page number for a log record 0
101208 17:17:45 mysqlbackup: Full backup completed!
```

```
# Back up some InnoDB tables and the .frm files for the backed-up tables only.
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --include='test\ib.*' \
  --only-innodb-with-frm=related backup
...many lines of output...
mysqlbackup: Scanned log up to lsn 2666737471.
mysqlbackup: Was able to parse the log up to lsn 2666737471.
mysqlbackup: Maximum page number for a log record 0
101208 17:17:45 mysqlbackup: Full backup completed!
```

バックアップディレクトリには、`ibstest` テーブルと `ibtest09` テーブルのバックアップだけが含まれます。ほかの InnoDB テーブルは `test\ib.*` のインクルードパターンに一致しませんでした。ただし、`ibtest11a`、`ibtest11b`、`ibtest11c`、`ibtest11d` のテーブルは、常にバックアップに含まれているシステムテーブルスペース (`ibdata1` file) に格納されているため、次に示すディレクトリに表示されていなくてもバックアップに含まれていることに注意してください。

```
# With the --only-innodb option:
$ ls /sqldata-backup/test
ibstest0.ibd  ibtest09.ibd
```

```
# With the --only-innodb-with-frm=related option:
$ ls /sqldata-backup/test
ibstest0.frm  ibtest09.frm
ibstest0.ibd  ibtest09.ibd
```

例 3.2 圧縮部分バックアップの作成

すべての InnoDB テーブルが独自のテーブルスペースを保持するように、MySQL を構成しました。`alex` または `blob` で始まる名前を持つ InnoDB テーブルだけを含む部分バックアップを作成します。`test` データベースのデータベースディレクトリの内容を次に示します。

```
$ ls /sqldata/mts/test
alex1.frm alex2.ibd blobt3.frm  ibstest0.ibd ibtest11a.frm ibtest11d.frm
alex1.ibd alex3.frm blobt3.ibd  ibtest09.frm ibtest11b.frm
alex2.frm alex3.ibd ibstest0.frm ibtest09.ibd ibtest11c.frm
```

`--compress` オプションと `--include` オプションを付けて `mysqlbackup` を実行します。

```
$ mysqlbackup --defaults-file=/home/pekka/.my.cnf --compress \
  --include='!(alex|blob).*' --only-innodb backup
```

```
...many lines of output...
mysqlbackup: Scanned log up to lsn 2666737471.
mysqlbackup: Was able to parse the log up to lsn 2666737471.
mysqlbackup: Maximum page number for a log record 0

mysqlbackup: Compressed 147 MB of data files to 15 MB (compression 89%).

101208 17:18:04 mysqlbackup: Full backup completed!
```

データベース `test` のバックアップディレクトリを次に示します。 `.ibz` ファイルは圧縮された per-table データファイルです。

```
$ ls /sqldata-backup/test
alex1.ibz alex2.ibz alex3.ibz blobt3.ibz
```

`mysqlbackup` コマンドの `--databases` オプションと `--databases-list-file` オプションを使用すれば、MySQL インスタンス全体ではなく選択したデータベースだけから、非 InnoDB テーブルをバックアップできます。(InnoDB テーブルをフィルタするには、代わりに `--include` オプションを使用します。)`--databases` では、データベース名のスペース区切りのリストを指定し、リスト全体を二重引用符で囲みます。`--databases-list-file` では、データベース名のリストを含むファイルのパスを、1 行に 1 つずつ指定します。

データベース名の一部またはすべてをテーブル名で修飾して、これらのデータベースから選択した非 InnoDB テーブルだけをバックアップできます。

このオプションを指定する場合は、必ず、すべてのバックアップ (基本的に増分バックアップ) で同じデータベースセットを含め、データベースの古いバージョンをリストアしないようにしてください。

3.3.5 単一ファイルバックアップの作成

多数のバックアップファイルの追跡および保護を回避し、バックアップデータの移動を簡略化するために、`mysqlbackup` コマンドでは、単一ファイル形式でのバックアップの作成、単一ファイルへの既存のバックアップのバック、元のバックアップディレクトリ構造への単一ファイルのアンバック、単一ファイルバックアップの内容の一覧表示、埋め込まれたチェックサムと照合した単一ファイルバックアップの内容の検証、または単一ファイルやディレクトリツリーの抽出を行えます。関連する `mysqlbackup` オプションの構文については、[セクション 5.1.10「単一ファイルバックアップオプション」](#) を参照してください。

単一ファイルバックアップは、テープバックアップや `scp` を始めとするコマンドなどの別のプロセスにストリーミングまたはパイプすることができるため、この手法を使用して、別のストレージデバイスまたはサーバーにバックアップを格納することができます。元のデータベースサーバーで大幅なストレージオーバーヘッドが発生することがなくなります。(単一ファイルバックアップの準備中、指定したバックアップディレクトリ内に、一時的に複数の小さな作業ファイルが用意されます。)

単一ファイルバックアップを作成するには、`mysqlbackup` のオプション `backup-to-image` を指定します。元のすべてのデータファイルは、異なるパスに分散させるのではなく、単一のディレクトリ下に置く必要があります。`datadir`、`innodb_log_group_home_dir`、および `innodb_data_home_dir` の構成オプションには同じパスを指定してください。

例 3.3 絶対パスへの単一ファイルバックアップ

このコマンドは、指定された絶対パスに単一のバックアップイメージを作成します。引き続き `--backup-dir` が必要で、一時的な出力、ステータス、およびメタデータファイルを保持するために使用されます。

```
mysqlbackup --backup-image=/backups/sales.mbi --backup-dir=/backup-tmp backup-to-image
```

例 3.4 相対パスへの単一ファイルバックアップ

このコマンドは、バックアップディレクトリより下の相対パスを付けて `--backup-image` を指定します。結果として生じる単一ファイルバックアップは、`/backups/sales.mbi` として作成されます。

```
mysqlbackup --backup-image=sales.mbi --backup-dir=/backups backup-to-image
```

例 3.5 標準出力への単一ファイルバックアップ

次のコマンドは、標準出力にバックアップ出力をダンプします。ここでも、`my.cnf` に指定された `--backup-dir` ディレクトリが一時ディレクトリとして使用されます。

```
mysqlbackup --backup-dir=/backups --backup-image=- backup-to-image > /backup/mybackup.mbi
```

例 3.6 単一イメージへの既存のバックアップディレクトリの変換

`my.cnf` に指定された `backup-dir` ディレクトリは、`/backup/my.mbi` ファイルにまとめられます。このディレクトリにはあらゆるものを格納でき、必ずしも MySQL Enterprise Backup で生成されたバックアップに限りません。

```
mysqlbackup --backup-image=/backup/my.mbi --backup-dir=/var/mysql/backup backup-dir-to-image
```

例 3.7 バックアップディレクトリへの既存のイメージの抽出

イメージの内容は、`backup-dir` にアンパックされます。

```
mysqlbackup --backup-dir=/var/backup --backup-image=/backup/my.mbi image-to-backup-dir
```

例 3.8 単一ファイルバックアップの内容の一覧表示

イメージの内容が一覧表示され、各行にファイルまたはディレクトリエントリが示されます。

```
mysqlbackup --backup-image=/backup/my.mbi list-image
```

例 3.9 現在のディレクトリへの単一ファイルバックアップの抽出

次のコマンドは、単一ファイルバックアップから現在の作業ディレクトリにすべての内容を抽出します。

```
mysqlbackup --backup-image=/var/my.mbi extract
```

例 3.10 バックアップディレクトリへの単一ファイルバックアップの抽出

このコマンドは、`image-to-backup-dir` オプションと同様に動作しますが、単一ファイルバックアップのすべての内容を `--backup-dir` ディレクトリに抽出します。

```
mysqlbackup --backup-image=/var/my.mbi --backup-dir=/var/backup extract
```

例 3.11 単一ファイルの選択的な抽出

次のコマンドは、単一ファイル `meta/comments.txt` をローカルパス `./meta/comments.txt` に抽出します。

```
mysqlbackup --backup-image=/var/my.mbi \
--src-entry=meta/comments.txt extract
```

次のコマンドは、`--dst-entry` オプションを使用することによって、`meta/comments.txt` ファイルを指定のパス `/tmp/mycomments.txt` に抽出します。

```
mysqlbackup --backup-image=/var/my.mbi \
--src-entry=meta/comments.txt \
--dst-entry=/tmp/mycomments.txt extract
```

次のコマンドは、`meta/comments.txt` (単一ファイルバックアップ内部) の内容を標準出力にダンプします。

```
mysqlbackup --backup-image=/var/my.mbi --src-entry=meta/comments.txt --dst-entry=- extract
```

例 3.12 単一ディレクトリの選択的な抽出

次のコマンドは、単一ディレクトリ `meta` をローカルファイルシステムパス `./meta` に抽出します。ディレクトリを抽出すると、サブディレクトリを含めそのすべての内容が抽出されます。

```
mysqlbackup --backup-image=/backup/my.mbi --src-entry=meta extract
```

次のコマンドは、`meta` ディレクトリのすべての内容 (そのすべてのファイルとサブディレクトリ) をディレクトリ `/tmp/my-meta` に抽出します。

```
mysqlbackup --backup-image=/backup/my.mbi --src-entry=meta \
--dst-entry=/tmp/my-meta extract
```

例 3.13 絶対パス名の処理

絶対パス名は、ローカルシステム内の同じパスに抽出されるため、そのパスの書き込みアクセス許可がない場合、問題が生じる可能性があります。次のようにして絶対パスを再マッピングできます。


```
mysqlbackup --backup-image=/backup/my.mbi --src-entry=/ --dst-entry=/myroot extract
mysqlbackup --backup-image=/backup/my.mbi --src-entry=. extract
```

最初のコマンドは、ローカルシステム内の `/myroot` ディレクトリへのすべての絶対パスを抽出します。2 番目のコマンドは、現在のディレクトリへのすべての相対パスを抽出します。

3.3.5.1 別のデバイスまたはサーバーへのバックアップデータのストリーミング

データベースサーバーでのストレージオーバーヘッドを制限するために、バックアップデータをローカルで格納せずに別のサーバーに転送できます。ストリーミングに関連した MySQL Enterprise Backup の主要な機能が、単一メッセージバックアップです。単一ファイルバックアップを標準出力に送信するには、`--backup-image` オプションを使用せず、`mysqlbackup` のオプション `backup-to-image` で指定します。(データが `stdout` に送信されていることを明らかにするために、`--backup-image=-` も指定できます。)データをストリーミングするには、標準出力から入力を取得してリモートシステムで同等のファイルを作成できる、パイプや `ssh/scp` などのオペレーティングシステム機能と組み合わせ、単一ファイルバックアップを使用します。単一ファイルバックアップをリモートシステムに直接格納するか、`image-to-backup-dir` オプションを付けて `mysqlbackup` コマンドを相手先で呼び出して、通常のバックアップのディレクトリ構造を再生できます。

例 3.14 リモートホストへの単一ファイルバックアップ

次のコマンドは、バックアップ出力をリモートホストにストリーミングし、出力はここで直接テープデバイスに保存できます。`--backup-dir=/tmp` は、最終の出力ファイルではなく、一時的な作業ファイルを格納するためのディレクトリを指定します。

```
mysqlbackup --backup-image=- --backup-dir=/tmp backup-to-image | \
ssh user@host command arg1 arg2...
```

単純にするため、すべての接続とほかの必要なオプションは、デフォルトの構成ファイルから取得される前提としています。リモートシステムで操作を実行できるように、`dd` や `tar` など、通常のアーカイブ手順の一環として使用するコマンドやデバイスなどの組み合わせを置き換えます。

例 3.15 リモートの MySQL Server への単一ファイルバックアップ

次のコマンドは、単一ファイルバックアップをリモートの MySQL Server にストリーミングします。

```
mysqlbackup --backup-dir=backup --backup-image=- --compress backup-to-image | \
ssh <user name>@<remote host name> 'mysqlbackup --backup-dir=backup_tmp --datadir=/data \
--innodb_log_group_home_dir= \
--innodb_log_files_in_group=<innodb_log_files_in_group_of_backedup_server> \
--innodb_log_file_size=<innodb_log_file_size_of_backedup_server> \
--innodb_data_file_path=<innodb_data_file_path_of_backedup_server> \
--uncompress --backup-image=- copy-back-and-apply-log'
```

3.3.5.2 テープへのバックアップ

テープドライブは、バックアップデータ用の価格が手頃な大容量のストレージデバイスです。MySQL Enterprise Backup 製品は、Oracle Secure Backup (OSB) などのメディア管理ソフトウェア (MMS) とやり取りして、MySQL バックアップおよびリストアジョブを進めます。メディア管理ソフトウェアは、バージョン 2 以上の System Backup to Tape (SBT) インタフェースをサポートしている必要があります。

Oracle Secure Backup などの MMS 製品と組み合わせたテープバックアップの実行の詳細は、[第9章「MySQL Enterprise Backup をメディア管理ソフトウェア \(MMS\) 製品と使用する」](#)を参照してください。

3.3.5.3 クラウドストレージへのバックアップ

MySQL Enterprise Backup は、バージョン 3.10.2 からクラウドバックアップをサポートしています。

クラウドストレージで作成でき、ここからリストアできるのは、単一ファイルバックアップだけです。単一ファイル操作と互換性のあるすべての `mysqlbackup` オプション (たとえば、`--incremental`、`--compression`、`--partial`、`--encryption` オプションなど) は、クラウドバックアップまたはリストアで使用できます。

現在、Amazon S3 が、MySQL Enterprise Backup でサポートしている唯一のクラウドサービスです。

クラウドバックアップは、`mysqlbackup` のクラウドオプションを使用して作成されます。これらのオプションの詳細は、[セクション5.1.15「クラウドストレージオプション」](#)に記載されています。次に、クラウドバックアップを作成するためのコマンド例を示します。

例 3.16 クラウドバックアップの作成

```
mysqlbackup\
--cloud-service=s3 --cloud-aws-region=<aws region> \
--cloud-access-key-id=<aws access key id> --cloud-secret-access-key=< aws secret access key> \
--cloud-bucket=<s3 bucket name> --cloud-object-key=<aws object key> \
--backup-dir=/home/user/dba/s3backuptmpdir \
--backup-image=- \
backup-to-image
```

`backup-to-image` に加え、単一ファイルバックアップに関するほかのすべての `mysqlbackup` 操作 (`backup-dir-to-image`、`list-image`、`validate`、`image-to-backup-dir`、`extract`、`copy-back`、および `copy-back-and-apply-log`) も、クラウドストレージを使用して実行できます。

例 3.17 クラウドストレージからバックアップディレクトリへの既存のイメージの抽出

`--backup-dir` オプションを使用してイメージの抽出先のディレクトリを指定して、クラウドストレージからバックアップイメージを抽出します。

```
mysqlbackup\
--cloud-service=s3 --cloud-aws-region=<aws region> \
--cloud-access-key-id=<aws access key id> --cloud-secret-access-key=< aws secret access key> \
--cloud-bucket=<s3 bucket name> --cloud-object-key=<aws object key> \
--backup-dir=/home/user/dba/s3backuptmpdir \
--backup-image=- \
image-to-backup-dir
```

クラウドストレージからバックアップイメージをリストアする方法については、[例4.9「クラウドストレージからMySQL Server への単一ファイルバックアップのリストア」](#)を参照してください。

3.3.6 オプティミスティックバックアップの作成

オプティミスティックバックアップは、頻繁に変更されるテーブルが少数しかない大規模なデータベースのバックアップおよびリストアのパフォーマンスを改善するために MySQL Enterprise Backup 3.11 で導入された機能です。

大規模なデータベース (テラバイト単位など) のホットバックアップ中、バックアップが進行しているときに、膨大な Redo ログファイルがサーバー上で生成される場合があります。Redo ログファイルが `mysqlbackup` で処理できる速度よりも速く増大すると、`mysqlbackup` が Redo ログサイクルに追いつけず、LSN が `mysqlbackup` によって読み取られる前にサーバーによって上書きされたときに、バックアップ操作は実際に失敗することがあります。さらに、`mysqlbackup` がバックアップに適用する巨大な `ibbackup_logfile` ファイル (大きな Redo ログファイルから作成) を持っているときに、[リストアできるようにバックアップを準備するための `apply-log` ステップ](#)に、非常に長い時間がかかることがあります。Redo ログの読み取りと書き込みに利用できる I/O リソースが、バックアップおよびリストアプロセス中に不足しているときに、問題は増大します。

オプティミスティックバックアップは、ユーザーには透過的である次の 2 つの内部フェーズにバックアッププロセスを分割することによって問題を軽減します。

1. **オプティミスティックフェーズ:** この最初のフェーズでは、バックアッププロセス中に変更される可能性の低いテーブル (下では「非アクティブテーブル」と呼ばれ、`optimistic-time` オプションや、除外によって `optimistic-busy-tables` オプションでユーザーが識別します) が、MySQL インスタンスをロックせずにバックアップされます。また、これらのテーブルは、バックアップが終了するまでに変更されることはないかと予想されるため、Redo ログ、Undo ログ、およびシステムテーブルスペースは、このフェーズでは `mysqlbackup` によってバックアップされません。
2. **通常フェーズ:** この 2 番目のフェーズでは、最初のフェーズでバックアップされていないテーブル (下では「ビジーテーブル」と呼ばれます) が、一般のバックアップで処理されるのと似た方法でバックアップされます。InnoDB ファイルが最初にコピーされ、続いて MySQL インスタンスがロックされた状態でほかの関連ファイルがコピーまたは処理されます。また、非アクティブテーブルの一部が、オプティミスティックフェーズでバックアップされたあとに、実際に変更されていることが判明した場合は、通常フェーズでもう一度コピーされます。Redo ログ、Undo ログ、およびシステムテーブルスペースもこのフェーズでバックアップされます。

オプティミスティックバックアップは、[optimistic-time](#) オプションまたは [optimistic-busy-tables](#) オプションが使用されているときに必ず行われます。これらのオプションの使用方法については、[セクション5.1.11「パフォーマンス/スケーラビリティ/容量オプション」](#)での詳細な説明を参照してください。予想どおりに、オプションで識別された非アクティブテーブルがバックアップ中にあまり変更されていない場合、[apply-log](#) 操作が非常に高速になるため、全体のバックアップ時間 (バックアップ操作の時間に加えて [apply-log](#) 操作の時間) は、通常のバックアップと比べ大幅に短縮される可能性があります。ただし、識別された非アクティブテーブルがバックアッププロセス中に大幅に変更されていることがわかった場合は、オプティミスティックバックアップの実行によるメリットは限られたものになり、最悪の場合、オプティミスティックバックアップの実行時間が実際に長くなることがあり、単一ファイルバックアップの場合は、バックアップのサイズが通常のバックアップと比べて増大します。したがって、ユーザーは、オプティミスティックバックアップを実行しようとするときには、どのテーブルが「非アクティブ」で、どのテーブルが「ビジー」かを慎重に識別する必要があります。

注記

オプティミスティックバックアップは、増分バックアップや、[トランスポータブルテーブルスペース \(TTS\)](#) を使用したバックアップには実行できません。

次の例で、オプティミスティックバックアップの作成方法を示します。

例 3.18 オプション [optimistic-time=YYMMDDHHMMSS](#) を使用したオプティミスティックバックアップ

この例では、2011年5月16日の正午以降に変更されたテーブルがビジーテーブルとして扱われ、オプティミスティックバックアップの通常フェーズでバックアップされます。その他のテーブルはすべてオプティミスティックフェーズでバックアップされます。

```
mysqlbackup --defaults-file=/etc/my.cnf --optimistic-time=110516120000 backup
```

例 3.19 オプション [optimistic-time=now](#) を使用したオプティミスティックバックアップ

この例では、すべてのテーブルが非アクティブテーブルとして扱われ、オプティミスティックバックアップのオプティミスティックフェーズでバックアップされます。

```
mysqlbackup --defaults-file=/etc/my.cnf --optimistic-time=now backup
```

例 3.20 [optimistic-busy-tables](#) オプションを使用したオプティミスティックバックアップ

この例では、名前に [mytables-](#) のサフィクスが付けられた [mydatabase](#) 内のテーブルがビジーテーブルとして扱われ、オプティミスティックバックアップの通常フェーズでバックアップされます。その他のテーブルはすべてオプティミスティックフェーズでバックアップされます。

```
mysqlbackup --defaults-file=/etc/my.cnf --optimistic-busy-tables='mydatabase\mytables-.*' backup
```

[optimistic-time](#) オプションと [optimistic-busy-tables](#) オプションの両方を使用し、これらがビジーテーブルにするテーブルの決定で競合した場合、[optimistic-busy-tables](#) が [optimistic-time](#) より優先されます。例:

例 3.21 [optimistic-busy-tables](#) オプションと [optimistic-time](#) オプションの両方を使用したオプティミスティックおよび部分バックアップ

この例では、名前に [mytables-](#) のサフィクスが付けられた [mydatabase](#) 内のテーブルがビジーテーブルとして扱われ、2010年5月16日の [optimistic-time](#) で指定された時間以降に変更されていない場合でも、通常フェーズでバックアップされます。

```
mysqlbackup --defaults-file=/etc/my.cnf --optimistic-busy-tables='mydatabase\mytables-.*' \
--optimistic-time=100516 backup
```

3.3.7 インメモリーデータベースデータのバックアップの作成

[mysqlbackup](#) コマンドの [--exec-when-locked](#) オプションを使用すると、データベースがまだロックされている間に、バックアップの終わり近くに実行するコマンドおよび引数を指定できます。このコマンドは、バックアップディレクトリで追加ファイルをコピーまたは作成できます。たとえば、このオプションを使用して、[mysqldump](#) コマンドで [MEMORY](#) テーブルをバックアップし、バックアップディレクトリに出力を格納できます。コマンドが実行するまでリダイレクトまたは変数置換を遅らせるには、パラメータ値全体を一重引用符で囲みます。

3.3.8 スケジュールされたバックアップの作成

定期的なバックアップスケジュールの維持は、MySQL Server にとってデータの損失を防止するための重要な手段です。このセクションでは、MySQL Enterprise Backup を実行するスケジュールを設定する簡単な複数の手段について説明します。

Linux およびほかの Unix のようなプラットフォーム: スケジュールされたバックアップについてシステムで cron ジョブを設定できます。2 つのタイプの cron ジョブがあります。特定のユーザーが所有し実行するユーザー cron ジョブを設定するには、次の手順に従います。

- MySQL Enterprise Backup を実行するユーザーとしてログオンし、次のコマンドを使用して、crontab を作成 (または変更) するためにエディタを呼び出します。

```
shell> crontab -e
```

- エディタで、次のようなエントリを crontab に追加し、変更内容を保存します。

```
@daily /path-to-mysqlbackup/mysqlbackup -uroot --backup-dir=/path-to-backup-folder/cronbackups --with-timestamp backup &>/dev/null
```

この crontab エントリは、毎日 00:00:00 に `mysqlbackup` を呼び出して `cronbackups` ディレクトリの下にバックアップを作成します。`stderr` と `stdout` ストリームからの出力は、`/dev/null/` にリダイレクトされるため、Cron サーバーの側でほかのアクション (たとえば、ユーザーへの通知の電子メール送信) を呼び出しません。

`root` が所有し実行するシステム cron ジョブを設定するには、`/etc/cron.d` フォルダの下にファイルを作成し、ユーザー (次の例の `root`) を `mysqlbackup` コマンドの前に追加して、このファイルを前述のような crontab エントリに入れます。

```
@daily root /path-to-mysqlbackup/mysqlbackup -uroot --backup-dir=/path-to-backup-folder/cronbackups --with-timestamp backup &>/dev/null
```

タイプの異なるスケジュールに応じて cron ジョブを設定するさまざまな方法の詳細は、プラットフォームのドキュメントを確認してください。

Windows プラットフォームの場合: タスクスケジューラを目的に合わせて使用します。手順については、Windows プラットフォームのドキュメントを確認してください。

第 4 章 データベースのリカバリとリストア

目次

4.1 バックアップをリストアする準備	43
4.2 リストア操作の実行	44
4.3 ホットバックアップからのポイントインタイム・リカバリ	46
4.4 単一の .ibd ファイルのバックアップとリストア	47
4.5 データベースのアップグレードまたはダウングレードによるバックアップのリストア	48

バックアップデータの最終的な目的は、データベースの問題からのリカバリに役立てることや、(通常、レポートクエリーを実行したり、新しいレプリケーションスレーブを作成したりするために) 別の場所に元のデータベースのクローンを作成することです。このセクションでは、さまざまなシナリオを処理する手順について説明します。

深刻なデータベースの問題が起きたあと、厳しい時間的な圧力下でリカバリを実行する必要性に迫られることがあります。前もって次の点を確認しておくことが重要です。

- データの転送やアンパックなどの処理を行うステップを含め、リカバリにかかる時間。
- 一度の試行で正しく行えるように、リカバリプロセスのすべてのステップについて実践ドキュメント化していること。ハードウェアの問題によって別のサーバーにデータをリストアする必要性が生じた場合は、そのサーバーでのすべての権限やストレージ容量などを前もって確認してください。
- リカバリ後、システムが適切に稼働するように、バックアップデータの正確さと完全さを定期的に検証していること。

4.1 バックアップをリストアする準備

バックアップの実行中にデータを挿入、更新、または削除できるため、バックアップジョブの完了直後は、バックアップファイルは一貫した状態でない可能性があります。これらの初期バックアップファイルは、**raw バックアップ**と呼ばれています。

特定の InnoDB **ログシーケンス番号**に対応するデータベースの状態を反映するように、バックアップファイルを更新する必要があります。(クラッシュリカバリと同じ種類の操作です。)このステップが完了すると、これらの最終的なファイルは、**準備されたバックアップ**と呼ばれます。

バックアップ中、**mysqlbackup** は、累積された InnoDB ログを **ibbackup_logfile** と呼ばれるファイルにコピーします。このログファイルは、バックアップしたデータファイルを「ロールフォワード」するために使用され、その結果、データファイル内のすべてのページが InnoDB ログの同じログシーケンス番号に対応します。このフェーズは、データファイルに対応する新しい **ib_logfiles** も作成します。

raw バックアップを準備されたバックアップに変換するための **mysqlbackup** オプションは、**apply-log** です。バックアップを行なった同じデータベースサーバー上でこのステップを実行するか、または最初に raw バックアップファイルを別のシステムに転送して、データベースサーバー上の CPU またはストレージオーバーヘッドを制限できます。

注記

apply-log 操作によってバックアップ内の元のファイルのいずれも変更されないため、操作が何らかの理由(ディスク領域の不足など)で失敗しても何も失われません。問題を修復したあと、安全に **apply-log** を再試行できます。**--force** オプションを指定することによって、失敗した **apply-log** 操作によって作成されたデータおよびログファイルを上書きできます。

単純なバックアップ(圧縮または増分バックアップを使用しないバックアップ)の場合、オプション **backup-and-apply-log** を使用して、初期バックアップと **apply-log** ステップを組み合わせることができます。

単一の **copy-back-and-apply-log** コマンドで **apply-log** と **copy-back**(準備されたバックアップをリストア)を実行することもできます。

例 4.1 バックアップへのログの適用

この例では、データをリストアできるようにするため、`mysqlbackup` を実行してデータファイルをロールフォワードします。

```
mysqlbackup --backup-dir=/export/backups/2011-06-21_8-36-58 apply-log
```

このコマンドは、バックアップディレクトリ内に InnoDB ログファイル (`ib_logfile*`) を作成し、ログレコードを InnoDB データファイル (`ibdata*` および `*.ibd`) に適用します。

例 4.2 圧縮バックアップへのログの適用

[セクション3.3.3「圧縮バックアップの作成」](#) に示したようにバックアップを圧縮した場合、バックアップにログを適用するときに、`mysqlbackup` に対して `--uncompress` オプションを指定します。

```
mysqlbackup --backup-dir=/export/backups/compressed --uncompress apply-log
```

例 4.3 完全バックアップへの増分バックアップの適用

[セクション3.3.2「増分バックアップの作成」](#) の説明に従って増分バックアップを取得したあとで、バイナリログから変更を適用する場合と同じ方法で、これらのバックアップファイルに反映された変更を完全バックアップに適用して、完全バックアップを最新の状態にする必要があります。

完全バックアップからのデータファイルを最新の状態にするには、完全バックアップが実行している間に行われたあらゆる変更がデータファイルに含まれるように、最初にログステップを適用します。続いて、増分バックアップからの変更内容を、完全バックアップで生成されたデータファイルに適用します。

```
mysqlbackup --backup-dir=/export/backups/full apply-log
mysqlbackup --backup-dir=/export/backups/full \
--incremental-backup-dir=/export/backups/incremental \
apply-incremental-backup
```

これで、`full-backup` ディレクトリ内のデータファイルは完全に、増分バックアップの時点の最新の状態になっています。

4.2 リストア操作の実行

リストア操作を実行するための `mysqlbackup` オプションは、`copy-back` と `copy-back-and-apply-log` です。リストアプロセスでは、データベースサーバーがすでにシャットダウンされていることが必要です (`--use-fts` オプションで作成したバックアップのリストアを除きます。下の [説明](#) を参照してください)。このプロセスは、データファイル、ログ、およびほかのバックアップされたファイルを、バックアップディレクトリからコピーしてその元の場所に戻し、必要な後処理をそれらに対して実行します。どのリストア操作の場合でも、オプション `datadir`、`innodb_log_files_in_group`、`innodb_log_file_size`、および `innodb_data_file_path` をターゲットサーバーの構成ファイル内、`--defaults-file` オプションによって指定されたファイル内、またはコマンド行オプションとして指定する必要があります。

例 4.4 データベースのシャットダウンとリストア

```
mysqladmin --user=root --password shutdown
mysqlbackup --defaults-file=/usr/local/mysql/my.cnf \
--backup-dir=/export/backups/full \
copy-back
```

注記

リストアしたデータには、`backup_history` テーブルが含まれ、MySQL Enterprise Backup はここに各バックアップの詳細を記録します。このテーブルを以前の状態にリストアすると、それ以降に行なったバックアップに関する情報が削除されます。これは、今後の増分バックアップの、特に `--incremental-base` オプションを使用した増分バックアップの正しい開始点です。

重要

完全リストアを実行するときに (たとえば、バックアップデータを新しい MySQL Server の設定に使用したり、既存の MySQL Server のすべてのデータの交換に使用したりするとき)、ターゲットデータディレクトリがすべてクリーンで、古いまたは不要なデータファイルが含まれていないことを確認してください。これには、`--datadir` オプションと `--innodb_data_file_path` オプションの両方で指定された場所で、ファイルを手動で削除することが必要になる可能性があります。同じクリーンアップは、`--use-tts` オプションで作成されたバックアップのリストアには必要なく (ただしこの場合、[Restoring Backups Created with the --use-tts Option](#) で示されたほかの要件は適用されます)、部分バックアップのリストアには通常必要ありません。

`copy-back-and-apply-log` オプションを代わりに使用することによって、`apply-log` 操作と `copy-back` 操作 (およびリストアしているバックアップの種類に応じて、多数のほかの操作も) を単一のステップに組み合わせることができます。

```
mysqlbackup --defaults-file=/usr/local/mysql/my.cnf \
--backup-dir=/export/backups/full \
copy-back-and-apply-log
```

例 4.5 圧縮バックアップのリストア

`copy-back-and-apply-log` を使用して、`<backupDir>` の圧縮バックアップを、サーバー上の `<restoreDir>` にリストアします。

```
mysqlbackup --defaults-file=<my.cnf> -uroot --backup-dir=<backupDir> --datadir=<restoreDir> \
--uncompress copy-back-and-apply-log
```

`--backup-dir` オプションを使用してイメージの抽出先の一時ディレクトリを指定して、`<image_name>` という名前の圧縮バックアップイメージに同じ操作を行います。

```
mysqlbackup --defaults-file=<backupDir>/backup-my.cnf -uroot --backup-image=<image_name> \
--backup-dir=<backupTmpDir> --datadir=<restoreDir> --uncompress copy-back-and-apply-log
```

圧縮バックアップの詳細は、[セクション3.3.3「圧縮バックアップの作成」](#) および [セクション5.1.7「圧縮オプション」](#) を参照してください。

例 4.6 暗号化されたバックアップイメージのリストア

`<keyFile>` という名前のファイルに含まれた暗号化鍵を使用して、`copy-back-and-apply-log` で `<image_name>` という名前の暗号化されたバックアップイメージをサーバー上の `<restoreDir>` にリストアします。

```
mysqlbackup --defaults-file=<backupDir>/backup-my.cnf --backup-image=<image_name> \
--backup-dir=<backupTmpDir> --datadir=<restoreDir> --decrypt --key-file=<keyFile> copy-back-and-apply-log
```

バックアップ暗号化と復号化の詳細は、[セクション5.1.14「暗号化オプション」](#) を参照してください。

例 4.7 増分バックアップイメージのリストア

`--incremental-backup-dir` オプションを使用してイメージの抽出先の一時ディレクトリを指定しながら、`copy-back-and-apply-log` で `<inc_image_name>` という名前の増分バックアップイメージをサーバー上の `<restoreDir>` にリストアします (増分バックアップの基準となっていた完全バックアップがすでに `<restoreDir>` にリストアされていることに注意してください)。

```
mysqlbackup --defaults-file=<backupDir>/backup-my.cnf -uroot --backup-image=<inc_image_name> \
--incremental-backup-dir=<incBackupTmpDir> --datadir=<restoreDir> --incremental \
copy-back-and-apply-log
```

`--backup-dir` を使用してイメージが抽出される一時ディレクトリを指定して同じことを行います。

```
mysqlbackup --defaults-file=<backupDir>/backup-my.cnf -uroot --backup-image=<inc_image_name> \
--backup-dir=<incBackupTmpDir> --datadir=<restoreDir> --incremental \
copy-back-and-apply-log
```

増分バックアップの詳細は、[セクション3.3.2「増分バックアップの作成」](#) および [セクション5.1.8「増分バックアップオプション」](#) を参照してください。

例 4.8 TTS バックアップからの選択されたテーブルのリストア

選択したテーブルは、`--include-tables` オプションと `--exclude-tables` オプションを使用して、**トランスポートブル テーブルスペース (TTS)** で作成されたバックアップ (つまり、`--use-tts` オプションで作成されたバックアップ) からリストアできます。次のコマンドは、バックアップから「sales」データベース内のすべてのテーブルをリストアしますが、「hardware」という名前のテーブルを除きます。

```
mysqlbackup --socket=/tmp/restoreserver.sock --datadir=/logs/restoreserverdata --backup-dir=/logs/backup \
--include-tables='^sales\.' --exclude-tables='^sales\.hardware$' copy-back-and-apply-log
```

TTS バックアップから選択したテーブルをリストアするときに適用される特殊な要件については、[Restoring Backups Created with the --use-tts Option](#)を参照してください。

例 4.9 クラウドストレージから MySQL Server への単一ファイルバックアップのリストア

`--backup-dir` オプションを使用してイメージの抽出先の一時ディレクトリを指定しながら、クラウドストレージからサーバー上の `datadir` にバックアップイメージをリストアします (クラウドサービスオプションの詳細は、[セクション 5.1.15 「クラウドストレージオプション」](#)を参照してください)。

```
mysqlbackup \
--defaults-file=/bkups/backupdir/backupmy.cnf \
--cloud-service=s3 --cloud-aws-region=<aws region> \
--cloud-access-key-id=<aws access key id> --cloud-secret-access-key=<aws secret access key> \
--cloud-bucket=<s3 bucket name> --cloud-object-key=<aws object key> \
--backup-dir=/home/user/dba/s3backuptmpdir \
--datadir=/home/user/dba/datadir \
--backup-image= \
copy-back-and-apply-log
```

--use-tts オプションで作成されたバックアップのリストア

`--use-tts` オプションで作成されたバックアップをリストアするには、複数の特殊な要件があります。

- 復元先のサーバーが実行中である必要があります。
- サーバーに接続するために必要なパラメータ (ポート番号、ソケット名など) が、`mysqlbackup` のコマンド行オプションとして与えられているか、またはデフォルトファイルの「`client`」セクションで指定されていることを確認します。
- バックアップが作成されたサーバーで使用されていた同じページサイズを、復元先のサーバーで使用している必要があります。
- 復元先のサーバーで `innodb_file_per_table` オプションを有効にする必要があります。
- リストアされるテーブルが、リストア先のサーバーに存在していることはできません。
- リストアされる per-table データファイル (`.ibd` ファイル) の InnoDB ファイル形式が、リストア先のサーバーでの `innodb_file_format` システム変数の値と一致しない場合、リストアが失敗します。この場合、形式には無関係に per-table データファイルをリストアできるようにするために、リストアコマンドとともに `--force` オプションを使用して、サーバーでの `innodb_file_format` の値を一時的に変更します。

4.3 ホットバックアップからのポイントインタイム・リカバリ

MySQL Enterprise Backup と、これによって作成される増分バックアップにデフォルトで含まれる**バイナリログファイル**を使用して、完全バックアップと増分バックアップの間や2つの増分バックアップの間の任意の時点の状態にデータベースをリストアできます。

データベースを特定の時点にリカバリするには:

1. バイナリロギングは、このリストア操作の基準として機能する完全バックアップを取得する前に、MySQL で有効にしておく必要があります。
2. データベースの最新の完全バックアップをリストアします。

- リストアしたバックアップの時点に対応するバイナリログの位置を探します。InnoDB は、トランザクションコミット時にテーブルスペースにバイナリログの位置情報だけを格納します。現在のバイナリロギングの位置を InnoDB に認識させるには、バイナリロギングが有効になっている間に、少なくとも 1 つのトランザクションを実行する必要があります。バックアップで `apply-log` 操作を実行すると、`mysqlbackup` によって、バックアップでわかっている最新の MySQL バイナリログの位置が出力されます。また、リストアされるデータで開始すると、`mysql` によってバイナリログの位置が出力されます。

```
$ mysql --defaults-file=/export/mysql/my.cnf
040122 15:41:57 InnoDB: Database was not shut down normally!
InnoDB: Starting crash recovery.
...
InnoDB: Last MySQL binlog file position 0 27183537, file name ./binlog.000005
...
mysql: ready for connections.
```

MySQL バージョンは 5.1 以上である必要があります。

出力された位置は、MySQL Enterprise Backup がデータファイルのコピーを終了した時点からの MySQL バイナリログのバイト位置です。

- `mysqlbinlog` を使用して、増分バックアップに含まれる **バイナリログ** ファイルから、バックアップのバイナリログの位置のあとに発生したすべての SQL アクティビティをダンプします。`mysqlbinlog` コマンドの出力を、`mysql` に直接パイプする代わりに、出力ファイルに送ります。この出力ファイルには、バイナリログ内のすべての SQL ステートメントのタイムスタンプが含まれます。

```
$ mysqlbinlog --start-position=27183537 /sqldata/binlog.000005 >partial_binlog
$ mysqlbinlog /sqldata/binlog.000006 >> partial_binlog
$ mysqlbinlog /sqldata/binlog.000007 >> partial_binlog
...
$ mysqlbinlog /sqldata/binlog.00000N >> partial_binlog
```

- エディタで、リストアする予定の特定の時点以降のすべてのステートメントを削除します。
- SQL ステートメントを再実行して、バックアップ後に行われた同じ操作でバックアップデータを更新します。次のように `mysql` で変更したファイル进行处理します。

```
$ mysql < partial_binlog
```

- データベースを可能な最新の時点にリカバリするには、ファイル内の `mysqlbinlog` 出力を保存し、そこから最新の SQL ステートメントを削除するステップをスキップします。`mysqlbinlog --start-position=...` からの出力を `mysql` に直接パイプして、最新のバックアップ後にすべての SQL ステートメントを再実行します。
- 2 つの増分バックアップの間の時点にデータベースをリストアするには、次の増分バックアップに含まれていたバイナリログファイルを使用して、ステップ 3 から 6 を繰り返します。

ポイントインタイム・リカバリでバイナリログを使用するためのヒントについては、[バイナリログを使用したポイントインタイム \(増分\) リカバリ](#) を参照してください。

4.4 単一の .ibd ファイルのバックアップとリストア

テーブル固有のテーブルスペースを持つテーブル (.ibd ファイルに格納) は、MySQL Server を停止させずに個別にリストアできます。間違ってテーブルデータを削除または更新した場合に、この手法を適用でき、`DROP TABLE`、`TRUNCATE TABLE`、または `DROP DATABASE` のステートメントを通じて、実際にはテーブル自体を失うことはありません。

.ibd ファイルのクリーンなバックアップがある場合、次のようにして、クリーンなバックアップが作成された MySQL インストール状態にリストアできます。

- MySQL 5.5 以前の場合、テーブルがすでに存在し、バックアップを取得して以降に破棄または切り捨てられていないことが必要です。InnoDB テーブルが切り捨てられるか削除されて再作成された場合、このテーブルは新しいテーブル ID を取得します。データベース内のテーブルとバックアップされたテーブルとの間で ID が一致しないと、このテーブルをリストアできないことがあります。テーブル ID が一致する必要があるのは、類似した一連のデータベースおよびテーブルを持つ別のサーバーではなく、バックアップデータが得られた同じ MySQL Server にリストアする必要がある理由でもあります。同じシリーズの MySQL Server の、ある一般提供開始 (GA) バージョンから別バージョンにリストアが行われるがぎり、この制限は MySQL 5.6 以降には適用されません。

2. リストアされるテーブルに対する書き込み操作を防止します。これにより、リストアの進行中にユーザーがテーブルを変更できなくなります。

```
LOCK TABLES tbl_name WRITE;
```

3. 次の `ALTER TABLE` ステートメントを発行します。

```
ALTER TABLE tbl_name DISCARD TABLESPACE;
```

注意: これにより現在の `.ibd` ファイルは削除されます。

4. バックアップ `.ibd` ファイルをコピーして、適切なデータベースディレクトリに戻します。
5. 次の `ALTER TABLE` ステートメントを発行します。

```
ALTER TABLE tbl_name IMPORT TABLESPACE;
```

6. 書き込みロックを解除して、リストア手順を完了します。

```
UNLOCK TABLES;
```

このコンテキストでは、クリーンな `.ibd` ファイルバックアップとは次のことを意味します。

- `.ibd` ファイル内に、トランザクションによってコミットされていない変更がない。
- `.ibd` ファイル内に、マージされていない挿入バッファエントリがない。
- ページにより、`.ibd` ファイルからすべての削除マークされたインデックスレコードが削除されている。
- `mysqld` が、`.ibd` ファイルのすべての変更されたページをバッファプールからファイルにフラッシュしている。

次の方法を利用して、このようなクリーンなバックアップ `.ibd` ファイルを作成できます。

1. `mysqld` サーバーからのすべてのアクティビティーを停止して、すべてのトランザクションをコミットします。
2. `SHOW INNODB STATUS` で、データベース内にアクティブなトランザクションがないことが表示され、`InnoDB` のメインスレッドステータスが `Waiting for server activity` になるまで待機します。続いて、`.ibd` ファイルのコピーを作成できます。

`.ibd` ファイルのクリーンなコピーを作成するもう 1 つの方法は、`mysqlbackup` を使用するというものです。

1. `--only-innodb` オプションまたは `--only-innodb-with-frm` オプションを付けて `mysqlbackup` を使用して、`InnoDB` インストールをバックアップします。
2. `mysqlbackup ... apply-log` を実行して、バックアップデータベースの一貫したバージョンを作成します。
3. 2 番目 (ダミー) の `mysqld` サーバーをバックアップ上で起動して、バックアップ内の `.ibd` ファイルをクリーンアップできるようにします。クリーンアップが終了するまで待機します。
4. ダミーの `mysqld` サーバーをシャットダウンします。
5. バックアップからクリーンな `.ibd` ファイルを取得します。

4.5 データベースのアップグレードまたはダウングレードによるバックアップのリストア

重要

MySQL 5.5 から 5.6 に移行した `InnoDB` ストレージエンジンに行われた変更のために、MySQL 5.5 データベースのバックアップを MySQL 5.6 Server にリストアするには、一般的なリストアおよびアップグレード手順以外の複数の追加ステップが必要になります。これを省略すると、ターゲットサーバーがクラッシュします。このようなリストアについては、次の [ステップ](#) に従ってください。

ある MySQL バージョンを実行しているサーバーをバックアップし、別の MySQL バージョンを実行しているサーバーでリストアできます。リストア後、新しい MySQL バージョンをはじめて実行する場合のように、適切なアップグレードステップを実行してください。(または古い MySQL を実行しているサーバーにインストールした場合は、適切なダウングレードステップを実行してください。)アップグレードとダウングレードの詳細は、[MySQL のアップグレード](#)と[MySQL のダウングレード](#)を参照してください。

注記

MySQL バージョンの特定の組み合わせの間でアップグレードしたあと、システムテーブルの定義の欠落または不一致に関するエラーメッセージが表示される場合があります。このような問題を修正するためのアップグレード手順で指示されているように、`mysql_upgrade` コマンドを使用します。このコマンドの説明については、[mysql_upgrade — MySQL テーブルのチェックとアップグレード](#)を参照してください。

警告

古い MySQL バージョンへのデータベースのリストア (つまり、サーバーのダウングレード) は、元のバージョンと最終バージョンが同じリリースシリーズ内に属しているとき (5.5.30 から 5.5.29 へなど) にもみサポートされます。より低いシリーズへのダウングレード (5.6.33 から 5.5.33 へなど) は、サーバークラッシュやデータの破損を引き起こす可能性があります。

MySQL 5.5 でバックアップし MySQL 5.6 でリストアするためのステップ

- MySQL 5.5 Server でデータをバックアップします。
- `apply-log` を実行し、続いてバックアップ上で `copy-back` 操作を実行して、MySQL 5.6 Server のデータディレクトリとして使用する予定のディレクトリにデータをリストアします。
- MySQL 5.6 Server 用のデータディレクトリを専用のデータディレクトリとして使用して、MySQL 5.5 Server を再起動します。

注記

これは、MySQL 5.5 のデータの MySQL 5.6 Server へのリストアに特有の追加ステップで、通常のリストアおよびアップグレード手順にはありません。MySQL 5.5 Server はこれを使用して、[クラッシュリカバリ](#)中にサーバーが行う処理に類似した複数のクリーンアップステップをデータに対して実行することにより、MySQL 5.6 へのアップグレード用にデータを準備します。

- MySQL 5.5 Server を停止させます。
- MySQL 5.6 Server をインストールします。
- MySQL 5.6 Server を起動します。
- リストアされるデータについて MySQL リファレンスマニュアルで記されているように、[アップグレードステップ](#)を実行します。MySQL 5.6 に付属している `mysql_upgrade` が適用されていることを確認します。
- データを確認します。

MySQL 5.1 でバックアップし MySQL 5.5 でリストアするためのステップ

- MySQL 5.1 でバックアップします。
- MySQL 5.5 をインストールします。
- MySQL 5.5 でリストアします。
- MySQL リファレンスマニュアルで記されているように、[アップグレードステップ](#)を実行します。
- データを確認します。

第 5 章 mysqlbackup コマンドリファレンス

目次

5.1 mysqlbackup コマンド行オプション	52
5.1.1 サブコマンド	57
5.1.2 標準オプション	63
5.1.3 接続オプション	64
5.1.4 サーバリポジトリオプション	65
5.1.5 バックアップリポジトリオプション	67
5.1.6 メタデータオプション	70
5.1.7 圧縮オプション	70
5.1.8 増分バックアップオプション	71
5.1.9 部分バックアップとリストアオプション	74
5.1.10 単一ファイルバックアップオプション	80
5.1.11 パフォーマンス/スケーラビリティ/容量オプション	82
5.1.12 メッセージロギングオプション	88
5.1.13 進行状況レポートオプション	89
5.1.14 暗号化オプション	92
5.1.15 クラウドストレージオプション	93
5.1.16 特別なバックアップの種類オプション	94
5.2 構成ファイルとパラメータ	95

`mysqlbackup` コマンドは、あらゆるバックアップおよびリストア操作のための使いやすいツールです。バックアップ操作時に、`mysqlbackup` は次をバックアップします。

- 次を含むすべての InnoDB テーブルおよびインデックス。
 - InnoDB **システムテーブルスペース**。これにはデフォルトですべての InnoDB テーブルが含まれます。
 - InnoDB **file-per-table** 設定で生成された個別のデータファイル。それぞれ、1 つのテーブルとそれに関連付けられたインデックスが含まれます。各データファイルは、元の **Antelope** または新しい **Barracuda** ファイル形式のいずれかを使用できます。
- すべての MyISAM テーブルおよびインデックス。
- ほかのストレージエンジンによって管理されるテーブル。
- 各テーブルの構造を記録する **.frm** ファイルなどの MySQL データディレクトリの下にあるその他のファイル。

バックアップの作成に加えて、`mysqlbackup` はバックアップデータをバックおよびアンバックし、バックアップ操作時に発生した InnoDB テーブルの変更をバックアップデータに適用して、データ、インデックス、およびログファイルを元の場所にリストアすることができます。

`mysqlbackup` を起動するサンプルコマンド行引数:

```
# Information about data files can be retrieved through the database connection.
# Specify connection options on the command line.
mysqlbackup --user=dba --password --port=3306 \
  --with-timestamp --backup-dir=/export/backups \
  backup

# Or we can include the above options in the configuration file
# under [mysqlbackup], and just specify the configuration file
# and the 'backup' operation.
mysqlbackup --defaults-file=/usr/local/mysql/my.cnf backup

# Or we can specify the configuration file as above, but
# override some of those options on the command line.
mysqlbackup --defaults-file=/usr/local/mysql/my.cnf \
  --compress --user=backupadmin --password --port=18080 \
```

backup

指定した `--user` と `--password` は MySQL サーバーに接続するために使用されます。この MySQL ユーザーには、[セクション3.1.2「MySQL の権限をバックアップ管理者に与える」](#)で説明するように、MySQL サーバーでの特定の権限が必要です。

`--with-timestamp` オプションは、上で指定したディレクトリの下に作成されるサブディレクトリにバックアップを配置します。バックアップサブディレクトリの名前は、バックアップが実行された日付と時間から形成されます。

ほかのコマンド行オプションの意味については、[セクション5.1「mysqlbackup コマンド行オプション」](#)を参照してください。構成パラメータについては、[セクション5.2「構成ファイルとパラメータ」](#)を参照してください。

`mysqlbackup` を実行するユーザーまたは cron ジョブが、MySQL データベースディレクトリからバックアップディレクトリへファイルをコピーする権限を持つことを確認してください。

バックアップが実行中の際、コマンドがサーバーへの接続を開いたまま維持できるように、接続タイムアウトの長さが十分であることを確認してください。`mysqlbackup` は、接続を有効なまま維持するために、各データベースのコピー後にサーバーを ping します。

重要

- `mysqlbackup` コマンドはデータベースの使用を妨げることなく、InnoDB テーブルをバックアップしますが、InnoDB 以外のファイル (MyISAM テーブルや `.frm` ファイルなど) をコピーする最終段階では、ステートメント `FLUSH TABLES WITH READ LOCK` を使用して、一時的にデータベースを読み取り専用状態にします。バックアップパフォーマンスを最高にし、データベース処理への影響を最小にするために:

1. バックアップの実行時に長い `SELECT` クエリーやその他の SQL ステートメントを実行しないでください。
2. MyISAM テーブルを比較的小さく維持し、おもに読み取り専用または読み取りが大半の作業用にします。

これにより、`mysqlbackup` 実行の最後のロックフェーズが短くなる (おそらく数秒) ため、`mysqld` の通常の処理をあまり妨げません。データベースアプリケーションで先述の条件が満たされない場合は、`--only-innodb` オプションを使用して、InnoDB テーブルのみをバックアップするか、または `--no-locking` オプションを使用して、InnoDB 以外のファイルをバックアップします。`--no-locking` 設定でコピーされる MyISAM、`.frm`、およびその他のファイルがバックアップの最終フェーズで更新されない場合、それらの整合性を保証できないことに注意してください。

- 大きなデータベースの場合、バックアップの実行に長時間かかることがあります。`mysqlbackup` コマンドが終了コード 0 を返していることを確認するか、または `mysqlbackup` がテキスト「`mysqlbackup completed OK!`」を出力していることを観察して、常に `mysqlbackup` が正常に完了していることをチェックしてください。
- `mysqlbackup` コマンドは、以前の MySQL 6.0 ソースツリーの「MySQL Backup」オープンソースプロジェクトとは異なります。MySQL Enterprise Backup プロジェクトは MySQL Backup イニシアチブに代わりました。
- テーブルに関する DDL 操作が実行中でない期間に、バックアップをスケジューリングしてください。DDL 操作と同時のバックアップに対する制約については、[セクション A.1「MySQL Enterprise Backup の制限」](#)を参照してください。

5.1 mysqlbackup コマンド行オプション

次のセクションでは、`mysqlbackup` の操作のさまざまなモードについて説明し、操作の各モードやさまざまな状況に適用可能なオプションについて詳しく説明します。

下の表に、`mysqlbackup` のすべてのコマンドオプションを示します。オプション名のホットリンクを使用して、オプションの詳細な説明にジャンプします。

注記

コマンドオプションは構成ファイルにも指定できます。セクション5.2「構成ファイルとパラメータ」の説明を参照してください。mysqlbackup コマンドは、構成ファイル、コマンド行、またはその両方に指定されているかどうかに関係なく、重複オプションを処理する場合に、MySQL の標準の方法に従います。オプションは最初に構成ファイルから処理され、次にコマンド行から処理されます。オプションが複数回指定されている場合、最後のインスタンスが優先されます。

表 5.1 すべてのオプションのリスト

オプション名	説明
<code>--backup-dir</code>	バックアップデータを格納するディレクトリ。
<code>--backup-image</code>	バックアップイメージのパス名を指定します。
<code>--backup_innodb_checksum_algorithm</code>	InnoDB テーブルスペースの検証に使用されるチェックサムアルゴリズムの名前。
<code>--backup_innodb_data_file_path</code>	バックアップ内の InnoDB システムのテーブルスペースファイルのパスとサイズを指定します。
<code>--backup_innodb_data_home_dir</code>	システムテーブルスペース内のすべての InnoDB データファイルのバックアップベースディレクトリ。
<code>--backup_innodb_log_file_size</code>	各 InnoDB バックアップログファイルのバイト単位でのサイズ。
<code>--backup_innodb_log_files_in_group</code>	バックアップ内の InnoDB ログファイルの数。
<code>--backup_innodb_log_group_home_dir</code>	InnoDB ログファイルのバックアップディレクトリ。
<code>--backup_innodb_page_size</code>	MySQL インスタンス内のすべての InnoDB テーブルスペースのページサイズ。
<code>--backup_innodb_undo_directory</code>	InnoDB が Undo ログの個別のテーブルスペースを作成する相対または絶対ディレクトリパス。
<code>--backup_innodb_undo_logs</code>	InnoDB がトランザクション内で使用するシステムテーブルスペース内のロールバックセグメントの数。
<code>--backup_innodb_undo_tablespace</code>	ゼロ以外の <code>innodb_undo_logs</code> 設定を使用する場合の Undo ログが分割されるテーブルスペースファイルの数。
<code>--character-sets-dir</code>	文字セットファイルのディレクトリ。
<code>--cloud-access-key-id</code>	Amazon S3 にログインするための AWS アクセス鍵 ID
<code>--cloud-aws-region</code>	mysqlbackup が S3 にアクセスする Amazon Web Services のリージョン
<code>--cloud-bucket</code>	バックアップイメージ用の Amazon S3 上のストレージバケット
<code>--cloud-object-key</code>	バックアップイメージ用の Amazon S3 オブジェクト鍵
<code>--cloud-proxy</code>	Amazon S3 にアクセスするために、環境のデフォルトのプロキシ設定をオーバーライドするプロキシアドレスとポート番号
<code>--cloud-secret-access-key</code>	AWS 秘密アクセス鍵
<code>--cloud-service</code>	データのバックアップまたはリストアのためのクラウドサービス (現在 Amazon S3 サービスの「s3」のみを受け付けます)
<code>--cloud-trace</code>	クラウド操作のトレース情報を出力します
<code>--comments</code>	コメント文字列を指定します。
<code>--comments-file</code>	コメントファイルのパスを指定します。
<code>--compress</code>	圧縮形式でバックアップを作成します。
<code>--compress-level</code>	圧縮のレベルを指定します。
<code>--compress-method</code>	圧縮アルゴリズムを指定します。
<code>--connect-if-online</code>	使用可能な場合にのみ接続を使用します。

オプション名	説明
<code>--connect_timeout</code>	秒単位での接続タイムアウト。
<code>--databases</code>	[レガシー] バックアップする InnoDB 以外のテーブルのリストを指定します。
<code>--databases-list-file</code>	[レガシー] バックアップする InnoDB 以外のテーブルを一覧表示するファイルのパス名を指定します。
<code>--datadir</code>	mysql サーバーデータディレクトリへのパス。
<code>--debug</code>	デバッグ情報を出力します。
<code>--decrypt</code>	MEB セキュアファイルに書き込まれたバックアップイメージを復号化します。
<code>--default-character-set</code>	デフォルトの文字セットを設定します。
<code>--defaults-extra-file</code>	グローバルファイルが読み取られた後にこのファイルを読み取ります。
<code>--defaults-file</code>	指定されたファイルからデフォルトのオプションのみを読み取ります。
<code>--defaults-group-suffix</code>	通常の名前と str のサフィクスが付いたオプショングループも読み取ります。
<code>--disable-manifest</code>	バックアップ操作のマニフェストファイルの生成を無効にします。
<code>--dst-entry</code>	単一ファイルバックアップで、ユーザー指定のパスに単一のファイルまたはディレクトリを抽出するために使用されます。
<code>--encrypt</code>	バックアップイメージを暗号化し、それを MEB セキュアファイルに書き込みます。
<code>--exclude-tables</code>	名前が正規表現 REGEXP に一致するバックアップテーブルから除外します。
<code>--exec-when-locked</code>	実行の終了近くにすべてのテーブルがロックされた場合、指定したユーティリティを除外します。
<code>--force</code>	操作に応じて、データ、ログ、またはイメージファイルの上書きを強制的に実行します。
<code>--help</code>	ヘルプを表示します。
<code>--host</code>	接続するホスト名。
<code>--include</code>	[レガシー] 正規表現 REGEXP に一致するテーブル単位の innodb データファイルのみをバックアップします。
<code>--include-tables</code>	正規表現 REGEXP に一致するバックアップテーブルに含めます。
<code>--incremental</code>	関連付けられた backup または backup-to-image 操作が増分であることを指定します。
<code>--incremental-backup-dir</code>	増分バックアップからデータを格納する場所を指定します。
<code>--incremental-base</code>	--incremental オプションのベースバックアップの指定。
<code>--incremental-with-redo-log-only</code>	バックアップに InnoDB データファイルを含めずに、InnoDB テーブルの増分バックアップを、Redo ログのバックアップへのコピーを基にするように指定します。
<code>--innodb_checksum_algorithm</code>	InnoDB テーブルスペースの検証に使用されるチェックサムアルゴリズムの名前。
<code>--innodb_data_file_path</code>	InnoDB システムのテーブルスペースファイルのパスとサイズを指定します。
<code>--innodb_data_home_dir</code>	共有システムのテーブルスペース内のすべての InnoDB データファイルのベースディレクトリを指定します。
<code>--innodb_log_file_size</code>	ロググループ内の各 InnoDB ログファイルのバイト単位でのサイズ。
<code>--innodb_log_files_in_group</code>	InnoDB ログファイルの数。
<code>--innodb_log_group_home_dir</code>	InnoDB ログファイルへのディレクトリパス。

オプション名	説明
<code>--innodb_page_size</code>	MySQL インスタンス内のすべての InnoDB テーブルスペースのページサイズ。
<code>--key</code>	暗号化と復号化に使用される対称鍵。
<code>--key-file</code>	暗号化と復号化に使用される対称鍵を含むファイルのパス名。
<code>--limit-memory</code>	MEB 操作に使用可能な M バイト単位でのメモリー。
<code>--log-bin-index</code>	使用されているすべてのバイナリログファイルを一覧表示する MySQL サーバー上のインデックスファイルの絶対パスを指定します (MySQL 5.5 およびすべてのオフラインバックアップの場合)。
<code>--login-path</code>	.mylogin.cnf ログインファイル内の指定したログインパスからオプションを読み取ります。
<code>--master-info-file</code>	スレーブがそのマスターに関する情報を記録する情報ファイルの絶対パスを指定します (スレーブサーバーのオフラインバックアップの場合のみ)。
<code>--messages-logdir</code>	メッセージログを格納するための既存のディレクトリのパス名を指定します。
<code>--no-connection</code>	サーバーに接続しません。
<code>--no-defaults</code>	指定されたファイルからデフォルトのオプションを読み取りません。
<code>--no-history-logging</code>	接続が使用できる場合でも、履歴ロギングを無効にします。
<code>--no-locking</code>	接続が使用できる場合でも、テーブルのロックを無効にします。
<code>--number-of-buffers</code>	バックアップ操作に使用するメモリーバッファの正確な数を指定します。
<code>--on-disk-full</code>	バックアッププロセスでディスク領域不足状況が検出された場合の動作を指定します。
<code>--only-innodb</code>	InnoDB データとログファイルのみをバックアップします。
<code>--only-innodb-with-frm</code>	[レガシー] InnoDB テーブルに関連付けられた InnoDB データ、ログファイル、および .frm ファイルのみをバックアップします。
<code>--only-known-file-types</code>	バックアップに既知の種類のリストのファイルのみを含めます。
<code>--optimistic-busy-tables</code>	オプティミスティックバックアップの最初のフェーズでスキップされるテーブルを選択するオプションで指定した正規表現を使用して、オプティミスティックバックアップを実行します。
<code>--optimistic-time</code>	オプティミスティック時間 (変更されていないテーブルが非アクティブテーブルとみなされるまでの時間) として、オプションで指定された値でオプティミスティックバックアップを実行します。
<code>--page-reread-count</code>	再読み取りする最大ページ数。
<code>--page-reread-time</code>	ページが再読み取りされるまでの待機時間。
<code>--password</code>	接続パスワード。
<code>--pipe</code>	--protocol=pipe のエイリアス
<code>--port</code>	接続先の TCP ポート番号。
<code>--print-defaults</code>	デフォルトのファイルで指定されているオプション値のリストを出力して、終了します。
<code>--process-threads</code>	バックアップ操作のプロセススレッドの数を指定します。
<code>--progress-interval</code>	秒単位での進行状況レポートの間隔。
<code>--protocol</code>	接続プロトコル。
<code>--read-threads</code>	バックアップ操作の読み取りスレッドの数を指定します。
<code>--relay-log-index</code>	使用されているすべてのリレーログファイルを一覧表示する MySQL サーバー上のインデックスファイルの絶対パスを指定します (スレーブサーバーのオフラインバックアップの場合のみ)。

オプション名	説明
<code>--relaylog-info-file</code>	スレーブがリレーログに関する情報を記録する情報ファイルの絶対パスを指定します (スレーブサーバーのオフラインバックアップの場合のみ)。
<code>--sbt-database-name</code>	テープバックアップのメディアとポリシーの選択について、メディア管理ソフトウェア (MMS) へのヒントとして使用されます。
<code>--sbt-environment</code>	SBT ライブラリに指定される環境変数割り当てのカンマ区切りリスト。
<code>--sbt-lib-path</code>	テープバックアップを管理するソフトウェアによって使用される SBT ライブラリのパス名。
<code>--secure-auth</code>	古い (4.1.1 以前) プロトコルを使用している場合に、サーバーへのクライアント接続を拒否します。
<code>--shared-memory-base-name</code>	Windows サーバーによって、クライアントの共有メモリーを使用した接続を許可するために使用される共有メモリー名を指定します (Windows のみ)
<code>--show-progress</code>	mysqlbackup に、その操作の進行状況インジケータと呼ばれる短い進行状況レポートを定期的に出力するように指示します。
<code>--skip-binlog</code>	バックアップにバイナリログファイルを含めません。
<code>--skip-messages-logdir</code>	teelog ファイルへのロギングを無効にします。
<code>--skip-relaylog</code>	バックアップにリレーログファイルを含めません。
<code>--skip-unused-pages</code>	InnoDB テーブルのバックアップ時に、テーブルスペース内の未使用ページをスキップします。
<code>--slave-info</code>	同一のスレーブサーバーのセットアップに必要な情報をキャプチャーします。
<code>--sleep</code>	1M バイトごとのデータをコピーした後にミリ秒単位でスリープする時間。
<code>--socket</code>	接続に使用するソケットファイル。
<code>--src-entry</code>	単一ファイルバックアップから抽出するファイルまたはディレクトリを識別します。
<code>--ssl</code>	接続に SSL を有効にします (ほかの <code>--ssl-</code> フラグによって自動的に有効にされます)。
<code>--ssl-ca</code>	PEM 形式の CA ファイル (<code>--ssl</code> を暗黙的に示します)。
<code>--ssl-capath</code>	CA ディレクトリ (OpenSSL ドキュメントを確認、 <code>--ssl</code> を暗黙的に示します)。
<code>--ssl-cert</code>	PEM 形式の X509 証明書 (<code>--ssl</code> を暗黙的に示します)。
<code>--ssl-cipher</code>	使用する SSL 暗号 (<code>--ssl</code> を暗黙的に示します)。
<code>--ssl-key</code>	PEM 形式の X509 鍵 (<code>--ssl</code> を暗黙的に示します)。
<code>--ssl-verify-server-cert</code>	接続時に使用されるホスト名に対して、証明書内のサーバーの「共通名」を確認します。
<code>--start-lsn</code>	前回のバックアップに含まれる最大の LSN 値を指定します。
<code>--suspend-at-end</code>	バックアップ手順が終了に近づいたときに、mysqlbackup コマンドを一時停止します。
<code>--trace</code>	mysqlbackup によるメッセージのトレースレベル。
<code>--uncompress</code>	ログ適用の前に圧縮バックアップを圧縮解除します。
<code>--use-tts</code>	トランスポータブルテーブルスペース (TTS) を使用した InnoDB テーブルの選択的バックアップを有効にします。
<code>--user</code>	接続するデータベースユーザー名。
<code>--verbose</code>	詳細情報を出力します。
<code>--version</code>	バージョン情報を表示します。

オプション名	説明
<code>--with-timestamp</code>	バックアップ操作のタイムスタンプから形成された名前で、バックアップディレクトリの下にサブディレクトリを作成します。
<code>--write-threads</code>	バックアップ操作の書き込みスレッドの数を指定します。

5.1.1 サブコマンド

これらのオプションは `mysqlbackup` コマンドのおもな操作またはモードを表します。 `mysqlbackup` の各呼び出しで指定できるのは 1 つだけであり、名前の前にダッシュは付けません。

これらのおもなオプションには、それぞれ固有の必須または使用可能なコマンドパラメータのセットがあります。たとえば、`backup*` オプションには、データベースサーバーへの接続情報が必要です。 `apply-log` およびバックアップデータの生成後にそれを操作するその他のオプションには、バックアップデータを配置する場所を指定するオプションが必要です。

サブコマンドのおもなグループ:

- バックアップ操作: `backup`、`backup-and-apply-log`、`backup-to-image`
- 適用操作: `apply-log`、`apply-incremental-backup`
- リストア操作: `copy-back`、`copy-back-and-apply-log`
- 検証操作: `validate`
- 単一ファイルバックアップ操作: `image-to-backup-dir`、`backup-dir-to-image`、`list-image`、`extract`

5.1.1.1 バックアップ操作

バックアップ操作は MySQL Enterprise Backup でもっとも頻繁に実行されるタスクです。圧縮バックアップや増分バックアップの `--compress` や `--incremental` を使用するなど、さまざまなオプションを追加することで、さまざまな種類のバックアップを実行できます。これはバックアップ操作を実行する場合の `mysqlbackup` コマンドの構文です。

```
mysqlbackup [STD-OPTIONS]
  [CONNECTION-OPTIONS]
  [SERVER-REPOSITORY-OPTIONS]
  [BACKUP-REPOSITORY-OPTIONS]
  [METADATA-OPTIONS]
  [COMPRESSION-OPTIONS]
  [SPECIAL-BACKUP-TYPES-OPTIONS]
  [INCREMENTAL-BACKUP-OPTIONS]
  [PARTIAL-BACKUP-RESTORE-OPTIONS]
  [SINGLE-FILE-BACKUP-OPTIONS]
  [PERFORMANCE-SCALABILITY-CAPACITY-OPTIONS]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  backup | backup-and-apply-log
```

```
mysqlbackup [STD-OPTIONS]
  [CONNECTION-OPTIONS]
  [SERVER-REPOSITORY-OPTIONS]
  [BACKUP-REPOSITORY-OPTIONS]
  [METADATA-OPTIONS]
  [COMPRESSION-OPTIONS]
  [SPECIAL-BACKUP-TYPES-OPTIONS]
  [INCREMENTAL-BACKUP-OPTIONS]
  [PARTIAL-BACKUP-RESTORE-OPTIONS]
  [SINGLE-FILE-BACKUP-OPTIONS]
  [PERFORMANCE-SCALABILITY-CAPACITY-OPTIONS]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  backup-to-image
```

- `backup`

バックアップの初期フェーズを実行します。後で、`apply-log` オプションを付けて `mysqlbackup` を再度実行し、第2フェーズを実行します。

- `backup-and-apply-log`

`backup` と `apply-log` の組み合わせ。増分バックアップと互換性がありません。さらに、`--compress` オプションとも互換性がありません。

- `backup-to-image`

バックアップファイルを保持するディレクトリ構造ではなく、単一ファイルのバックアップを生成します。宛先ファイルを指定する `--backup-image` オプションが必要です。データをデータベースサーバーに保存することなく、バックアップをストレージデバイスまたは別のシステムにストリーミングするために使用できます。標準出力を表す `--backup-image=-` を指定して、出力を別のコマンドにパイプすることができます。通常の情報メッセージがバックアップ出力と混ざらないように、`--help` メッセージ、エラー、アラート、および通常の情報メッセージは、常に標準エラーに出力されます。

例 5.1 デフォルトの構成ファイルからの接続パラメータによる簡単なバックアップ

次の例に、`mysqlbackup` コマンドで、デフォルトの MySQL 構成ファイルの「`mysqlbackup`」セクションのデータベースに必要な接続パラメータを使用した最小バックアップを示します。

```
mysqlbackup --backup-dir=/export/backups/latest backup
```

例 5.2 基本的な増分バックアップ

```
mysqlbackup --incremental --start-lsn=12345 --incremental-backup-dir=/path/to/incbackup backup
```

増分バックアップ専用の個別のディレクトリがあります。このディレクトリと、フルバックアップ用のディレクトリのどちらも `my.cnf` ファイルに指定でき、バックアップの種類に応じて、適切なディレクトリが使用されます。正常なリストア操作の実行には、増分バックアップデータと以前のフルバックアップデータのどちらも必要です。

5.1.1.2 既存のバックアップデータの `apply-log` 操作

これらの操作では、バックアップの進行中に発生した InnoDB テーブルへのすべての変更を含めて、バックアップファイルを最新にします。便宜上、`backup-and-apply-log` オプションを使用して、この操作と初期バックアップを組み合わせることができますが、増分バックアップまたは圧縮バックアップを実行する場合は、手順を個別に実行する必要があります。

```
mysqlbackup [STD-OPTIONS]
  [--limit-memory=MB] [--uncompress] [--backup-dir=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  apply-log
```

```
mysqlbackup [STD-OPTIONS]
  [--incremental-backup-dir=PATH] [--backup-dir=PATH]
  [--limit-memory=MB] [--uncompress]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  apply-incremental-backup
```

- `apply-log`

バックアップを実行中のデータへのすべての変更を含め、バックアップ内の InnoDB テーブルを最新にします。

- `apply-incremental-backup`

増分バックアップからのデータを使用して、バックアップを最新にします。

例 5.3 フルバックアップへのログの適用

```
mysqlbackup --backup-dir=/path/to/backup apply-log
```

`backup-dir` 内の `backup-my.cnf` ファイルを読み取り、バックアップを理解します。`my.cnf` デフォルトファイルは、`apply-log` 操作を実行中のメモリーの使用を制限する `limit-memory=MB` 値を指定する以外の効果はありません。

apply-log 操作は増分バックアップに適用されないため、この操作に `incremental-backup-dir` は必要ありません。

単一の `copy-back-and-apply-log` コマンドによって `apply-log` と `copy-back` (準備されたバックアップをリストアする) をまとめて実行することもできます。

5.1.1.3 既存のバックアップのリストア

バックアップからデータファイルをデータベースサーバー内のそれらの元の場所にリストアします。リストア操作の前に、まず MySQL インスタンスをシャットダウンする必要があります。オプション `datadir`、`innodb_log_files_in_group`、および `innodb_log_file_size` をターゲットサーバーの構成ファイル内、`--defaults-file` オプションによって指定されたファイル内、またはコマンド行オプションとしてのいずれかに指定する必要があります。使用方法と例については、[第4章「データベースのリカバリとリストア」](#)を参照してください。

```
mysqlbackup [STD-OPTIONS]
  [SERVER-REPOSITORY-OPTIONS]
  [--backup-dir=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PARTIAL-BACKUP-RESTORE-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  copy-back

mysqlbackup [STD-OPTIONS]
  [SERVER-REPOSITORY-OPTIONS]
  [--backup-image=IMAGE]
  [--backup-dir=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PARTIAL-BACKUP-RESTORE-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  copy-back-and-apply-log
```

- `copy-back`

バックアップから MySQL サーバー内の元の場所にファイルをリストアします。

完全なリストアを実行する (バックアップデータを使用して、新しい MySQL サーバーをセットアップしたり、既存の MySQL サーバーのすべてのデータを置換したりする場合など) 前に、リストア先のディレクトリへの何らかのクリーンアップ作業が必要になることがあります。詳細は、[セクション4.2「リストア操作の実行」](#) [45]を参照してください。

`--use-tts` オプションを使用して作成したバックアップをリストアする場合は、いくつかの特別な要件があります。詳しくは「[Restoring Backups Created with the --use-tts Option](#)」を参照してください。

- `copy-back-and-apply-log`

1ステップで、`--backup-image` オプションで指定した単一ファイルバックアップまたは `--backup-dir` オプションで指定したディレクトリからのバックアップをサーバーのデータディレクトリにリストアし、`apply-log` 操作を実行して、リストアしたデータを最新にします。単一ファイルバックアップをリストアするための複数ステップのアプローチ (これは一般に、圧縮されたイメージをリストアする場合の `extract`、`uncompress`、`apply-log`、および `copy-back`、または圧縮されていないイメージの場合の `extract`、`apply-log`、および `copy-back` の連続した手順の実行から構成される) と比較して、このオプションにより、リストアプロセスが簡単かつ高速になり、必要なディスク領域も節約されます。

次に、`copy-back-and-apply-log` を使用したさまざまな種類のバックアップリストアの特別な要件をいくつか示します。

- 圧縮されたディレクトリやイメージをリストアするには、コマンド行に `--uncompress` オプションを含めます。
- 単一ファイルバックアップをリストアするには、`--backup-image` オプションでバックアップイメージの場所を指定するほか、`--backup-dir` オプションで、リストアプロセス中に生成される一時ファイルを格納するために使用されるフォルダの場所も指定します。
- 増分バックアップディレクトリをリストアするには、フルバックアップ (増分バックアップの基となる) がすでにリストアされていることを前提とします。

- コマンド行に `--incremental` オプションを含めます。
- `--backup-dir` または `--incremental-backup-dir` オプションを使用して、増分バックアップディレクトリを指定します。
- 単一ファイルの増分バックアップをリストアするには、`--backup-image` オプションで増分バックアップイメージの場所を指定するほか、`--backup-dir` または `--incremental-backup-dir` オプションで、リストアッププロセス中に生成される一時ファイルを格納するために使用されるフォルダの場所も指定します。
- `--use-tts` オプションで作成されたバックアップをリストアするには:
 - 「[Restoring Backups Created with the --use-tts Option](#)」で説明している一般的な要件を参照してください。
 - オプション設定 `--use-tts=with-minimum-locking` で作成されたイメージバックアップをリストアする場合、`--backup-dir` オプションで、バックアップ内のすべてのテーブルを一時的に抽出するためと、`apply-log` 操作を実行して、データをサーバーのデータディレクトリにリストアする前に、データを最新にするために使用されるフォルダの場所を指定します。
 - オプション `--use-tts` で作成されたバックアップディレクトリをリストアする場合、バックアップディレクトリに対して `apply-log` 操作が実行されます。これは、取得されたバックアップがプロセス中に変更されることを意味するため、ユーザーは、何らかの問題が発生した場合にバックアップデータの損失を避けるため、リストアを続行する前に、バックアップディレクトリの余分なコピーを作成した方がよいと考えられます。

次にも注意してください。

- `--skip-unused-pages` オプションによって作成されたバックアップは、`copy-back-and-apply-log` を使用してリストアできません。
- MySQL Enterprise Backup 3.8.2 以前で取得されたイメージバックアップの場合、バックアップ内の `.isl` ファイルによって指し示されているテーブルごとの `.ibd` ファイルは、`copy-back-and-apply-log` によって、`.isl` ファイルで指し示されている場所ではなく、サーバーのデータディレクトリにリストアされます。

`copy-back-and-apply-log` 操作の最後に、データディレクトリ内にファイル `backup_variables.txt` が作成されるか、更新されます。このファイルには、リストアされた内容に関するメタデータが格納され、その後の増分バックアップの単一ステップのリストアによって使用されます。ユーザーが削除または変更しないでください。

`copy-back-and-apply-log` サブコマンドでさまざまな種類のバックアップをリストアするサンプルコマンドについては、[セクション4.2「リストア操作の実行」](#)を参照してください。

警告

レプリケーション目的でサーバーをリストアする際に、バックアップされたサーバーで `innodb_undo_directory` オプションを使用して、Undo ログをデータディレクトリ以外に配置した場合、`copy-back` または `copy-back-and-apply-log` で、`--defaults-file` オプションに対してファイル `server-my.cnf` または `server-all.cnf` を使用する際は、ファイル内に `innodb_undo_directory` オプションを正しく構成するように注意してください。そうしないと、元のサーバー上のデータやログファイルが誤って上書きされることがあります。

5.1.1.4 バックアップの検証

バックアップデータの完全性を確保するため、MySQL Enterprise Backup では、バックアップが作成されるか、別のシステムに転送された後に、そのデータページのチェックサム値によって、バックアップを検証するための `validate` サブコマンドを提供しています。

```
mysqlbackup [STD-OPTIONS]
  [--backup-dir=PATH][--backup-image=IMAGE]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  validate
```

- `validate`

バックアップが破損または切り捨てられていないことを確認します。この操作は、バックアップ内の各データページのチェックサム値を検証します。

著しく破損しているファイルで、過剰な時間やリソースを費やすことを避けるため、`mysqlbackup` は `.ibd` ファイル内に 20 を超える破損したページが見つかったら、そのファイルの検証を停止し、次のファイルに進みます。その場合、操作のサマリーに、すべての破損ページ数が示されず、「20 ページ以上破損している」ことのみが示されません。

この操作には次の制限もあります。

- すべてのバックアップディレクトリで、この操作では、その中の InnoDB データファイル (`ibdata*` ファイルと `*.ibd` ファイル) のみを検証できます。バックアップディレクトリ内のほかのファイルの種類の問題 (`.frm` ファイルの破損など) は検出されません。
- `.ibd` ファイルまたは `.frm` ファイルが、バックアップ時にバックアップディレクトリ内にないか、またはバックアップの作成後にバックアップディレクトリから削除されている場合、`validate` 操作で問題を検出できません。
- その中の `.ibd` ファイルから、ページが削除されるか、切り捨てられることによって、バックアップディレクトリが破損している場合、`validate` 操作で問題を検出できません。

これはバックアップディレクトリを検証する場合のサンプルコマンドです。

```
mysqlbackup -uroot --backup-dir=/logs/backupext validate
```

これはバックアップイメージを検証する場合のサンプルコマンドです。

```
mysqlbackup -uroot --backup-image=/logs/fullimage.mi validate
```

次に、暗号化されたバックアップイメージを検証するためのサンプルコマンドと、成功した検証の出力を示します。

```
$ mysqlbackup --backup-image=/meb/backups/image.mbi --decrypt --key-file=/meb/enckeyfile validate
```

```
140219 11:22:44 mysqlbackup: INFO: Validating image ... /logs/img.bi
140219 11:22:44 mysqlbackup: INFO: Validate: [Dir]: meta
140219 11:22:45 mysqlbackup: INFO: Total files as specified in image: 44
mysqlbackup: INFO: datadir/tpch/tabnorm7.ibd Validated...
mysqlbackup: INFO: datadir/tpch/tabnorm8.ibd Validated...
mysqlbackup: INFO: datadir/tpch/tabnorm9.ibd Validated...
.....
140219 11:22:45 mysqlbackup: INFO: Validate operation completed successfully.
140219 11:22:45 mysqlbackup: INFO: Backup Image validation successful.
mysqlbackup: INFO: Source Image Path = /logs/img.bi
mysqlbackup completed OK!
```

これはヘッダーのチェックサムの不一致のサンプル出力です。

```
mysqlbackup: ERROR: Checksum mismatch.
Computed checksum: ###      Checksum in image: ### mysqlbackup: ERROR: Problem verifying checksum of initial header of backup image.
Image Path = /meb/backups/image.mbi
mysqlbackup: ERROR: Backup image validation failed.
```

これは、破損した `.ibd` ファイルを含むイメージのサンプル出力です。

```
mysqlbackup: ERROR: datadir/db2/bigtab1.ibd has corrupt page number : 64  page number from page header : 64
mysqlbackup: ERROR: datadir/db2/bigtab1.ibd is corrupt and has : 10 corrupt pages
mysqlbackup: ERROR: datadir/db2/t1.ibd has corrupt page number : 4  page number from page header : 0
.....
mysqlbackup: ERROR: datadir/db2/t1.ibd is corrupt and has : 5 corrupt pages
mysqlbackup: ERROR: datadir/ibdata1 has corrupt page number : 63  page number from page header : 63
mysqlbackup: ERROR: datadir/ibdata1 has corrupt page number : 7  page number from page header : 7
.....
mysqlbackup: ERROR: datadir/ibdata1 is corrupt and has : 10 corrupt pages
mysqlbackup failed with errors!
```

これは圧縮されたバックアップディレクトリの成功した検証のサンプル出力です。

```
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabnorm5.ibz Validated...
```

```
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabnorm6.ibz Validated...
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabnorm7.ibz Validated...
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabnorm8.ibz Validated...
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabnorm9.ibz Validated...
mysqlbackup: INFO: /backups/backup-dir/datadir/tpch/tabrowformat.ibz Validated...
140219 11:22:45 mysqlbackup: INFO: Validate backup directory operation completed successfully.
```

5.1.1.5 単一ファイルバックアップの操作

バックアップデータの転送と管理を簡単にするため、各バックアップを単一のファイル（バックアップイメージ）に維持することができます。`backup-to-image` オプションで単一のファイルに対して直接バックアップを実行したり、またはこのオプションで既存のバックアップを単一のファイルにパックしたり、単一ファイルバックアップをフルバックアップディレクトリ構造にアンパックしたりすることができます。単一ファイルバックアップを操作するためのオプションはほかにもあり、それらについて下に説明します。使用方法と例については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

```
mysqlbackup [STD-OPTIONS]
  [--backup-image=IMAGE] [--backup-dir=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  image-to-backup-dir

mysqlbackup [STD-OPTIONS]
  [--backup-dir=PATH] [--backup-image=IMAGE]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  backup-dir-to-image

mysqlbackup [STD-OPTIONS]
  [--backup-image=IMAGE] [--src-entry=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  list-image

mysqlbackup [STD-OPTIONS]
  [--backup-image=IMAGE]
  [--backup-dir=PATH]
  [--src-entry=PATH] [--dst-entry=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  extract

mysqlbackup [STD-OPTIONS]
  [SERVER-REPOSITORY-OPTIONS]
  [--backup-image=IMAGE]
  [--backup-dir=PATH]
  [MESSAGE-LOGGING-OPTIONS]
  [PARTIAL-BACKUP-RESTORE-OPTIONS]
  [PROGRESS-REPORT-OPTIONS]
  [ENCRYPTION-OPTIONS]
  [CLOUD-STORAGE-OPTIONS]
  copy-back-and-apply-log
```

- [image-to-backup-dir](#)

単一ファイルバックアップをフルバックアップディレクトリ構造にアンパックします。イメージファイルとアンパック先のディレクトリの両方のパスを指定します。使用方法と例については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

- [backup-dir-to-image](#)

既存のバックアップを単一のファイルにパックします。-（標準出力）の `--backup-image` 値を指定して、既存のバックアップディレクトリ構造をテープデバイス、またはバックアップを別のサーバーに転送するコマンドにストリー

ミングします。`--backup-image` パラメータは - または `backup-dir` ディレクトリ以外の絶対パスのいずれかです。使用方法と例については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

- `list-image`

単一ファイルバックアップの内容を表示します。イメージ内のすべてのファイルとディレクトリを一覧表示します。`--src-entry=name` は特定のファイルまたはディレクトリを一覧表示するために使用できます。名前がディレクトリである場合、イメージ内のそのすべてのファイルとサブディレクトリが再帰的に一覧表示されます。使用方法と例については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

注記

`list-image` 操作は、クラウドプロキシで Range ヘッダーをサポートしている場合にのみ、クラウドバックアップに対して実行できます。

- `extract`

単一ファイルバックアップから個々のファイルまたはディレクトリをアンパックします。バックアップデータの完全なセットを必要としないトラブルシューティングまたはリストア操作。結果のファイルやディレクトリは、現在のディレクトリ、または指定されている場合は `backup-dir` に入れます。絶対パス名を持つイメージ内のすべてのファイルとディレクトリの内容は、ローカルシステム上の同じ絶対パス名に抽出されます。使用方法と例については、[セクション3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

`--src-entry=path` オプションは、イメージ内の単一ファイルまたは単一ディレクトリを選択して抽出するために使用できます。イメージ内に存在しているとおり、パスを指定します。

`--dst-entry=path` オプションは `--src-entry=path` オプションとともに、単一のファイルまたは単一のディレクトリをユーザー指定のファイルやディレクトリにそれぞれ抽出するために使用できます。`--src-entry` オプションを使用するが、`--dst-entry` オプションを省略した場合、選択したファイルやディレクトリがローカルファイルシステム内の同じパスに抽出されます。

デフォルトの抽出先は現在の作業ディレクトリです。それは、`--backup-dir` オプションによって上書きできます。イメージ内の相対パスを持つすべてのファイルは、宛先ディレクトリの相対パス名に抽出されます。

イメージに絶対パス名を持つ一部のエントリが含まれる場合、それらのエントリは、`--backup-dir` オプションが指定されている場合でも、同じ絶対パス名に抽出されます。絶対パス名の場所を変更するには、`--dst-entry` オプションを使用する必要があります。

- `copy-back-and-apply-log`

[セクション5.1.1.3「既存のバックアップのリストア」](#)の `copy-back-and-apply-log` の説明を参照してください。

5.1.2 標準オプション

標準オプションは、一般的な性質を持つオプションまたは、ほかの特定のオプショングループに分類されないオプションです。

次に、標準オプションのリストを示します。

- 次の標準オプションは、`mysql` コマンドにも存在します。これらのオプションの完全な説明については、「[サーバーオプションおよび変数リファレンス](#)」などからアクセス可能な MySQL リファレンスマニュアルを参照してください。これらのオプションは、残りの標準オプションを含めて、ほかのすべての `mysqlbackup` オプションの前に指定する必要があります。

```
--print-defaults  Print the program argument list and exit.
--no-defaults     Don't read default options from any option file.
--defaults-file=PATH  Only read default options from the given file. It has to be the first option to be specified, if used.
--defaults-extra-file=PATH  Read this file after the global files are read.
--defaults-group-suffix=str  Also read option groups with the usual names and a suffix of str.
```

- さらに次のオプションは、`mysqlbackup` と `mysql` で共通しており、それらの完全な説明については、「[サーバーオプションおよび変数リファレンス](#)」からアクセス可能な MySQL リファレンスマニュアルを参照してください。ただし、`mysqlbackup` では、`mysql` と異なり、これらのオプションの短縮形を受け付けません (たとえば、`mysqlbackup` では、`-h` の代わりに `--help` を使用する必要があります)。

```
--help    Display help.
--version Display version information
--verbose Print more verbose information.
--debug   Print debug information.
```

- `mysqlbackup` ではより多くの標準オプションを使用できます。

```
--force Force operations such as overwrite files, create backup directory.
--trace=level Trace level of messages by mysqlbackup.
```

`--force`: デフォルトで、一部の操作では、既存のファイルに書き込むように伝えたと、ユーザーデータやログファイルを上書きせずに、停止します。`--force` により、`apply-log` および `apply-incremental-backup` 操作時に、InnoDB データおよびログファイルの上書きが可能になり、`backup-to-image` または `backup-dir-to-image` オプションの使用時にイメージファイルの置換が可能になります。ほかのすべての操作では、`--force` オプションはエラーメッセージを伴って拒否されます。

`--trace=level`

コマンド行形式	<code>--trace=LEVEL</code>
型	列挙
デフォルト	0
有効な値	0 1 2 3

`mysqlbackup` メッセージのトレースレベル。詳細度が増す順番での許容レベル:

- 0 - INFO (情報、警告、エラー)
- 1 - FINE (詳細オプションが有効)
- 2 - FINER (デバッグオプションが有効)
- 3 - FINEST (すべての低レベル出力を含む)

5.1.3 接続オプション

`mysqlbackup` はバックアップを作成する際に、データベース接続を使用して、SQL コマンドを MySQL サーバーに送信します。一般的な接続の詳細は、MySQL リファレンスマニュアルの「MySQL サーバーへの接続」の説明と同じです。

`mysqlbackup` 呼び出しの一部として、MySQL サーバーへの接続に必要な適切な `--user`、`--password`、`--port`、および `--socket` オプションを指定します。

MySQL 構成ファイルの「`mysqlbackup`」または「`client`」セクション内または `mysqlbackup` コマンド行オプションから、次の接続固有のオプションを指定できます。`mysqlbackup` はデフォルトの構成ファイルを読み取り、次にコマンド行に指定された `my.cnf` ファイルを読み取ります。

注記

- `mysqlbackup` は「`client`」グループから `--user`、`--password`、`--port`、および `--socket` オプションのみを読み取り、ほかのすべての接続オプションを無視します。
- `--password` の値を指定しないと、コマンドによってキーボードからの入力が必要になります。
- `--host` オプションは、互換性のため構成ファイルで使用できますが、現在無効です。`mysqlbackup` コマンドは常にローカルサーバーの IP アドレスに接続します。

```
Options Common to mysqld
=====

--login-path=name
--port=port-num
--protocol=tcp|socket|pipe|memory
--pipe [ alias for --protocol=pipe ]
--user=name [ short option: -u ]
--host=hostname
--socket=name
--shared-memory-base-name=value [Windows only]
--character-sets-dir=PATH
--default-character-set=VALUE
--secure-auth [ Don't connect to pre-4.1.1 server ]
--password[=value] [ short option: -p ]
--connect_timeout
--ssl [ Enable SSL for connection ]
--ssl-key=file_name
--ssl-cert=file_name
--ssl-ca=file_name
--ssl-capath=directory_name
--ssl-cipher=cipher_list
--ssl-verify-server-cert

Connection Options Specific to mysqlbackup
=====

--no-connection
--connect-if-online
```

mysql コマンドによって使用されるほかのほとんどの接続パラメータは認識されますが、通知なく無視されます。不明な接続パラメータによって、mysqlbackup コマンドは停止します。

次の接続オプションは mysqlbackup に固有です。

- `--no-connection`

`--no-connection` オプションは、ほかの接続オプションより優先され、ファイルレベルの操作を使用して、バックアップを実行します。このオプションを使用する場合、値が通常データベース接続経由で自動的に取得される多くのオプションを、構成ファイル内またはコマンド行に指定する必要があります。

警告

このオプションは `--no-history-logging` オプションと `--no-locking` オプションも有効にするため、これにより、バックアップ操作中にテーブルが変更された場合に、InnoDB 以外のデータに不整合が発生する可能性があります。さらに、これは後続の増分バックアップにも影響することがあります。詳細については、`--incremental-base` オプションの説明を参照してください。

- `--connect-if-online`

デフォルトで、データベース接続は、初期段階でソースリポジトリ構成を取得するためと InnoDB 以外のデータのコピー中にテーブルをロックするための両方のバックアップ操作に使用されます。このオプションにより、mysqlbackup は両方のフェーズで接続を試行しますが、接続を確立できない場合でも続行します。接続を確立できない場合、処理は `--no-connection` オプションを使用した場合と同じになります。このオプションは、バックアップ操作時にデータベースサーバーが停止した場合など、緊急の状況で役立つことがあります。

5.1.4 サーバーリポジトリオプション

これらのリポジトリオプションは、データのバックアップ元やバックアップのリストア先など、データベースサーバーに関連する各種パラメータを指定します。

これらのオプションは、次の操作でのみ使用します。

- バックアップ作成操作: `backup`、`backup-and-apply-log`、`backup-to-image`。
- リストア操作: `copy-back`、`copy-back-and-apply-log`。

バックアップ時にデータベース接続を使用できる場合、ソースリポジトリを表すパラメータは無視され、データベース接続から取得された対応する値によって上書きされます。

MySQL サーバーに対してこれらのオプションを指定する方法については、オプション名をクリックして、MySQL リファレンスマニュアルの説明を参照してください。

- `datadir=PATH`

これは、MySQL インスタンスで使用される `datadir` 値です。`.frm` ファイルはインスタンス内のデータベースにちなんで名付けられたサブディレクトリ内のここに存在します。

データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。これは MySQL サーバーと MySQL Enterprise Backup の両方にきわめて重要なパラメータです。

- `innodb_data_home_dir=PATH`

InnoDB データファイルが存在するディレクトリを指定します。通常 `datadir` と同じですが、別にもすることもできます。

このパラメータは、`innodb_data_file_path=SIZE` とともに、MySQL サーバー内の `ibdata1`、`ibdata2` などの InnoDB データファイルが置かれる場所を決定します。

一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。

その値は次のように取得されます。

- `innodb_data_home_dir` を指定しない場合、`datadir` の値が継承されます。
- `innodb_data_home_dir` が相対パスの場合、そのパスは `datadir` 値に相対的 (つまりその下) に配置されます。
- "" の `innodb_data_home_dir` は / ルートディレクトリを表します。
- `innodb_data_home_dir` が絶対パスの場合、その値はそのまま使用されます。

- `innodb_data_file_path=VALUE`

InnoDB データファイルの名前とサイズを指定します。例:

```
ibdata1:32M;ibdata2:32M:autoextend
/abs/path/ibdata1:32M:autoextend
innodb-dir/ibdata1:32M:autoextend
```

データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。

このパラメータは `innodb_data_home_dir` とともに、サーバーリポジトリ内の InnoDB データファイル (`ibdata1`、`ibdata2` など) が存在する場所を指定します。

一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。データベース接続を使用できない場合、それを指定する必要があります。

初期ファイル名が / 文字で始まっているかどうかに関係なく、ファイルは `innodb_data_home_dir` 値に相対的に配置されます。

- `innodb_log_group_home_dir=PATH`

サーバーリポジトリ内の InnoDB ログが存在する場所を指定します。通常 `datadir` と同じですが、別にもすることもできます。

その値は次のように取得されます。

- `innodb_log_group_home_dir` を指定しない場合、`datadir` の値が継承されます。
- `innodb_log_group_home_dir` が相対パスの場合、そのパスは `datadir` 値に相対的 (つまりその下) に配置されます。

- `innodb_log_group_home_dir` が絶対パスの場合、その値はそのまま使用されます。
- `innodb_log_files_in_group=N`
ローテーションされる前の InnoDB ログファイル数を指定します。
一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。データベース接続を使用できない場合、それを指定する必要があります。
データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。
- `innodb_log_file_size=SIZE`
次のログファイルに切り替えるまでの単一の InnoDB ログファイルの最大サイズを指定します。例: 20M。
一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。データベース接続を使用できない場合、それを指定する必要があります。
データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。
- `innodb_page_size=SIZE`
すべての InnoDB テーブルスペースのページサイズを指定します。
一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。データベース接続を使用できない場合、それを指定する必要があります。
データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。
- `innodb_checksum_algorithm=NAME`
InnoDB テーブルスペースの検証に使用されるチェックサムアルゴリズムの名前を指定します。デフォルトは `innodb` です。
一般に、値はデータベース接続を使用して自動的に取得されるため、このオプションを指定する必要はありません。データベース接続を使用できない場合、それを指定する必要があります。
データベース接続が存在している場合、値は自動的に取得され、指定した値はすべてオーバーライドされます。

5.1.5 バックアップリポジトリオプション

これらのオプションは、バックアップディレクトリのレイアウトに関連する各種パラメータを指定します。

`--backup-dir` オプションは、このグループからユーザーが一般に指定する唯一のオプションです。すべての `backup_innodb_*` オプションの値は、`backup_` プリフィクスのない対応する構成オプションから自動的に取得できます (下の各オプションの説明を参照してください)。これらのオプションは、バックアップ時に、ユーザーがバックアップのリストア時に使用される、それらのオプションの特定の値を指定する必要がある場合にのみ指定する必要があります。`backup_innodb_*` オプションで指定された値は `backup_my.cnf` ファイルに格納され、将来のリストア時に使用されます。

バックアップ時にデータベース接続を使用できる場合、それらの `backup_innodb_*` オプションは無視され、それらの値がデータベース接続から取得された対応する値によって上書きされます。

これらのオプションは、次の操作でのみ使用します。

- バックアップ作成操作: `backup`、`backup-and-apply-log`、`backup-to-image`。
- リストア操作: `copy-back`、`copy-back-and-apply-log`。

次のパラメータはバックアップディレクトリ内のファイルのレイアウトを示します。

- `--backup_dir=PATH`

`--backup-dir` と同じです。バックアップデータを格納するディレクトリ。これはほとんどの種類のバックアップ操作に必要なきわめて重要なパラメータです。これは `--datadir` で指定されるディレクトリのサブディレクトリにする

ことはできません。追加のレベルのサブディレクトリは、`--with-timestamp` オプションも指定した場合に作成されません。

さらに、`copy-back-and-apply-log` で単一ファイルバックアップをリストアする場合、リストアプロセス時に生成される一時ファイルの格納に使用されるフォルダの場所を指定するために `--backup-dir` が使用されます。

- `backup_innodb_data_home_dir=PATH`

バックアップ InnoDB データファイルが存在するディレクトリを指定します。通常 `backup-dir` と同じですが、別にもすることもできます。

このパラメータは、`backup_innodb_data_file_path` とともに、バックアップディレクトリ構造内の InnoDB データファイル (`ibdata1`、`ibdata2` など) が格納される場所を指定します。

このパラメータはバックアップ操作のみに適用され、リストアには適用されません。

バックアップ操作 (`backup`、`backup-and-apply-log`、`backup-to-image` など) の場合、バックアップ先ディレクトリの値は次のように取得されます。

- `backup_innodb_data_home_dir` を指定しない場合、`backup-dir` の値が継承されます。
- `backup_innodb_data_home_dir` が相対パスの場合、そのパスは `backup-dir` 値に相対的 (つまりその下) に配置されます。
- "" の `backup_innodb_data_home_dir` は / ルートディレクトリを表します。
- `backup_innodb_data_home_dir` が絶対パスの場合、その値はそのまま使用されます。

バックアップディレクトリを簡単に再配置できるようにし、`backup-my.cnf` ファイルの編集を避けるため、バックアップ操作では、この値が `backup-dir` 値と異なる場合にのみ、可能な限り相対パスを使用して、`backup-my.cnf` に書き込みます。

`backup-to-image` 操作の場合、単一ファイルバックアップがマシンに依存しないように、`backup_innodb_data_home_dir` オプションの最終値を相対パスにする必要があります。

- `backup_innodb_data_file_path=VALUE`

InnoDB データファイルの名前とサイズを指定します。例:

```
ibdata1:32M;ibdata2:32M:autoextend
/abs/path/ibdata1:32M:autoextend
innodb-dir/ibdata1:32M:autoextend
```

このパラメータは `backup_innodb_data_home_dir` とともに、バックアップリポジトリ内の InnoDB データファイル (`ibdata1`、`ibdata2` など) が存在する場所を指定します。

バックアップディレクトリ内で、相対パスで指定されたすべてのデータファイルは `backup_dir` パスに相対的に配置されます。絶対パスで指定されたすべてのデータファイルは、`backup_innodb_data_home` ディレクトリ内に配置されます。

パラメータが指定されていない場合、`innodb_data_file_path` オプションの値から値を継承します。ソースと宛先の両方で同じファイルに解決される絶対パスを使おうとすると、バックアップは取り消されます。

バックアップ内の InnoDB データファイルに絶対パスを指定するには、`backup_innodb_data_home` オプションを "" に設定する必要もあります。

- `backup_innodb_log_group_home_dir=PATH`

バックアップ InnoDB ログが存在する場所を指定します。通常 `backup-dir` と同じですが、別にすることもできます。

ログファイルの名前は修正されますが、再構成できません。

このパラメータはバックアップ操作のみに適用されます (リストアには適用されません)。

バックアップ操作はこの値を使用して、`backup-my.cnf` に `innodb_log_group_home_dir=value` としてそれを書き込みます。

`copy-back` および `apply-log` 操作の場合、`backup-my.cnf` 内の `innodb_log_group_home_dir` はその作成方法と互換性があるように処理されます。

- `backup_innodb_log_files_in_group=N`

ローテーションされるまでのバックアップ内の InnoDB ログファイルの数を指定します。例: 5。

通常 `innodb_log_files_in_group` と同じですが、別にすることもできます。

このパラメータの値は次として取得されます。

- コマンド行または構成ファイルから指定された `backup_innodb_log_files_in_group` 値。
- または使用可能な場合にデータベース接続からの `innodb_log_files_in_group` 値。
- またはコマンド行または構成ファイルからの `innodb_log_files_in_group` 値。

- `backup_innodb_log_file_size=SIZE`

次のログファイルに切り替えるまでのバックアップ内の単一の InnoDB ログファイルの最大サイズを指定します。例: 20M。

通常 `innodb_log_file_size` と同じですが、別にすることもできます。

このパラメータの値は次として取得されます。

- コマンド行または構成ファイルからの指定された `backup_innodb_log_file_size` 値。
- または使用可能な場合にデータベース接続からの `innodb_log_file_size` 値。
- またはコマンド行または構成ファイルからの、指定された `innodb_log_file_size` 値。

- `backup_innodb_page_size=SIZE`

MySQL インスタンス内のすべての InnoDB テーブルスペースのページサイズ。ページサイズは MySQL インスタンスと同じ値にする必要があります。デフォルトで、サーバーの `innodb` ページサイズを想定します。

- `backup_innodb_checksum_algorithm=NAME`

InnoDB テーブルスペースの検証に使用されるチェックサムアルゴリズムの名前。デフォルトは「`innodb`」です。

- `backup_innodb_undo_directory=PATH`

InnoDB が Undo ログの個別のテーブルスペースを作成する相対または絶対ディレクトリパス。一般に別のストレージデバイスにそれらのログを配置するために使用します。

- `backup_innodb_undo_logs=NUMBER`

InnoDB がトランザクション内で使用するシステムテーブルスペース内のロールバックセグメントの数。この設定は、Undo ログに関連する相互排他競合が観察された場合のパフォーマンスのチューニングに適切です。

- `backup_innodb_undo_tablespaces=NUMBER`

ゼロ以外の `innodb_undo_logs` 設定を使用する場合の Undo ログが分割されるテーブルスペースファイル数。デフォルトで、すべての Undo ログはシステムテーブルスペースの一部であり、システムテーブルスペースには常に `innodb_undo_tablespaces` によって構成されたものに加えて、1 つの Undo テーブルスペースが含まれます。

- `--with-timestamp`

バックアップ操作のタイムスタンプから形成された名前で、バックアップディレクトリの下にサブディレクトリを作成します。多くのバックアップスナップショットを格納する単一のバックアップディレクトリを保守するために便利です。

デフォルト: タイムスタンプされたサブディレクトリは作成されません。新しいバックアップに同じバックアップディレクトリを再利用するには、前のバックアップファイルを手動で削除するか、`--force` オプションを指定して、それらを上書きします。

5.1.6 メタデータオプション

これらのオプションは、バックアップに関するメタデータの生成を制御します。メタデータはバックアップディレクトリに保存されるものと、バックアップされたインスタンスの `mysql` データベース内のテーブルに保存されるものがあります。

- `--no-history-logging`

バックアップの進行状況と履歴の、バックアップされたデータベース内のロギングテーブルへの記録をオフにします。これらのテーブルの詳細については、[セクション 10.4 「MySQL Enterprise Backup ログの使用」](#) を参照してください。

デフォルト: 履歴ロギングは有効です。`--no-connection` を指定した場合、履歴ロギングは自動的に無効にされます。`--connect-if-online` を指定した場合、バックアップ時にデータベース接続が正常に確立された場合にのみ履歴ロギングが機能します。

- `--comments=STRING`

コマンド行形式	<code>--comments=STRING</code>
型	文字列

バックアップについて説明したり、識別したりするコメント文字列を指定します。複数語のコメントは適切な引用符で囲みます。この文字列はバックアップ内のファイル `meta/comments.txt` に保存されます。例: `--comments="Backup of HR data on 2010/12/10"`。

- `--comments-file=PATH`

コマンド行形式	<code>--comments-file=PATH</code>
型	ファイル名

バックアップについて説明するコメントを格納するファイルへのパスを指定します。このファイルはバックアップに `meta/comments.txt` という名前で保存されます。例: `--comments-file=/path/to/comments.txt`。

このオプションは両方が指定されている場合に、`--comments` オプションをオーバーライドします。

5.1.7 圧縮オプション

バックアップの圧縮の概要については、[セクション 3.3.3 「圧縮バックアップの作成」](#) を参照してください。

- `--compress`

圧縮形式でバックアップを作成します。通常のバックアップの場合、InnoDB データファイルのみが圧縮され、圧縮後、それらには `.ibz` 拡張子が付けられます。同様に、単一イメージバックアップでは、バックアップイメージ内の InnoDB データファイルのみが圧縮されます。

デフォルト: 圧縮は無効です。

- `--compress-method=ALGORITHM`

コマンド行形式	<code>--compress-method=ALGORITHM</code>
型	列挙
デフォルト	lz4
有効な値	zlib lz4 lzma

圧縮アルゴリズムを指定します。オプションのサポートされる引数と、それらが表すアルゴリズム:

- **lz4**: LZ4 r109。サポートされる 3 つのアルゴリズムのうち、これはもっとも効率的で、一般に最小の CPU コストで、バックアップとリストアにかかる時間が最短になります。ほかの圧縮アルゴリズムとの比較を含む詳細については、[lz4-きわめて高速な圧縮アルゴリズム](#)を参照してください。
- **lzma**: LZMA 9.20。サポートされる 3 つのアルゴリズムのうち、これは一般に最高の圧縮率を提供しますが、CPU コストの点でもほかの 2 つのオプションよりかなり高価になります。そのため、これはアクティブなシステムにはお勧めしませんが、時間外または非アクティブなデータベースまたは I/O レートがきわめて低い場所でのみお勧めします。
- **zlib**: ZLIB v1.2.3。これは、速度と圧縮率の点で、ほかの 2 つのサポートされるアルゴリズムの間にあります。ZLIB は、3.10 以前の MySQL Enterprise Backup バージョンで使用できる唯一の圧縮アルゴリズムです。

デフォルト: lz4。構成ファイルまたはコマンド行からオプションの値を明示的に指定すると、`--compress` オプションが自動的に有効になります。

- `--compress-level=LEVEL`

コマンド行形式	<code>--compress-level=LEVEL</code>
型	数値
デフォルト	1
最小値	0
最大値	9

「0」から「9」の範囲の圧縮のレベルを指定します。「0」は圧縮を無効にし、「1」は最高速の圧縮で、「9」は最大(および最低速)の圧縮になります。オプションは、ZLIB または LZMA アルゴリズムを使用した圧縮にのみ意味があります。`--compress-method` オプションによってほかのアルゴリズムが選択された場合、それは無視されます。

デフォルト: 1 (最低で最高速の圧縮)。構成ファイルまたはコマンド行から 0 以外の値を明示的に指定すると、`--compress` オプションが自動的に有効になります。

- `--uncompress`

`apply-log` または `copy-back-and-apply-log` 操作で使用した場合、InnoDB ログを適用する前に、圧縮バックアップを圧縮解除します。

5.1.8 増分バックアップオプション

増分バックアップの概要とこれらのオプションに関する使用情報については、[セクション3.3.2「増分バックアップの作成」](#)を参照してください。

増分バックアップを取得するには、`--incremental-backup-dir` と一緒に、`--incremental` または `--incremental-with-redo-log-only` を指定します。指定した LSN 後に変更されたすべての InnoDB が増分バックアップにコピーされます。`--incremental` または `--incremental-with-redo-log-only` の選択に応じて、ほかのオプションが必要であったり、推奨されたりします。

- `--incremental`

関連付けられた `backup` または `backup-to-image` 操作が増分であることを指定します。さらに、`--incremental-base` オプションまたは `--start-lsn` オプションと `--incremental-backup-dir` オプションの組み合わせのいずれかが必要です。

増分の観点は InnoDB テーブルにのみ適用されます。デフォルトで、すべての InnoDB 以外のファイルと `.frm` ファイルも増分バックアップに含まれます。増分バックアップで InnoDB 以外のデータを除外するには、`--only-innodb` オプションを使用します。

- `--incremental-with-redo-log-only`

`backup` または `backup-to-image` 操作の増分バックアップの代替形式を指定します。さらに、`--incremental-base` オプションまたは `--start-lsn` オプションと `--incremental-backup-dir` オプションの組み合わせのいずれかが必要です。

このオプションによって実行される増分バックアップには、`--incremental` オプションと異なるパフォーマンス特性と操作上の制限があります。

- InnoDB テーブルへの変更は、InnoDB Redo ログの内容に基づいて判断されます。Redo ログファイルは、事前にわかっている固定のサイズを持つため、データベースのサイズ、DML アクティビティーの量、Redo ログファイルのサイズによっては、変更されたページを見つけるために、InnoDB のテーブルスペースファイルをスキャンするよりも、それらからの変更を読み取る方が必要な I/O が少なくなる可能性があります。
- Redo ログファイルは循環バッファーとして動作し、新しい DML 操作が行われると、古い変更の記録が書ききられるため、ワークロードに対して生成されるログファイルのサイズと Redo データの量に依存する予測可能なスケジュールで、新しい増分バックアップを取得する必要があります。そうしないと、Redo ログは前回の増分バックアップからのすべての変更を記録するために十分なだけさかのぼらないことがあります。この場合、`mysqlbackup` コマンドはすぐに続行できないことを判断し、エラーを返します。バックアップスクリプトは、エラーをキャッチでき、代わりに `--incremental` オプションで増分バックアップを実行します。

例:

- Redo ログのサイズを計算するには、コマンド `SHOW VARIABLES LIKE 'innodb_log_file%'` を発行し、出力に基づいて、`innodb_log_file_size` 設定と `innodb_log_files_in_group` を乗算します。物理レベルで Redo ログサイズを計算するには、MySQL インスタンスの `datadir` ディレクトリを調べ、パターン `ib_logfile*` に一致するファイルのサイズを合計します。
- InnoDB LSN 値は Redo ログに書き込まれるバイト数に対応します。特定の時点で LSN をチェックするには、コマンド `SHOW ENGINE INNODB STATUS` を発行し、LOG 見出しの下を調べます。バックアップ戦略を計画する場合、LSN 値を定期的に記録し、現在の値から以前の値を引いて、毎時、毎日などに生成される Redo データの量を計算します。

MySQL 5.5 以前では、MySQL サーバーが正常にシャットダウンせずに強制終了した場合に、起動時間が長くなるのを避けるため、Redo ログをかなり小さく維持することが一般的な方法でした。MySQL 5.5 以降では、「InnoDB 構成変数の最適化」に説明するように、クラッシュリカバリのパフォーマンスが大幅に向上しています。それらのリリースでは、Redo ログファイルを大きくすることが、バックアップ戦略やデータベースワークロードに役立つ場合に、そうすることができます。

- この増分バックアップの種類は、標準 `--incremental` オプションとして、小さすぎる `--start-lsn` 値を許容しません。たとえば、フルバックアップを作成できず、その後すべて同じ `--start-lsn` 値を使用する一連の `--incremental-with-redo-log-only` バックアップを作成します。次の増分バックアップの開始 LSN として、前回のバックアップの正確な終了 LSN を指定してください。任意の値を使用しないでください。

注記

このオプションを使用して、連続した増分バックアップ間で、LSN 値を正確に一致させるため、Oracle では、`--incremental-with-redo-log-only` オプションを使用する場合に常に `--incremental-base` オプションを使用することをお勧めします。

- 特定の MySQL インスタンスに対し、この増分バックアップの種類が現実的であり、効率的であるか判断するには:
 - InnoDB Redo ログファイル内のデータの変更の速度を測定します。LSN を定期的にチェックし、一定の時間や日数の間に累積する Redo データの量を確認します。
 - Redo ログの累積率と Redo ログファイルのサイズを比較します。この率を使用して、増分バックアップを取得する頻度を確認し、Redo ログで履歴データを使用できないために、バックアップが失敗する可能性を避けます。たとえば、1日あたり 1G バイトの Redo ログデータを生成し、Redo ログファイルの合計サイズが 7G バイトである場合、増分バックアップを週 1 回以上の頻度にスケジュールします。突然の更新の急増によって、通常より多くの Redo が生成された場合の可能性のある問題を避けるため、毎日または 2 日おきに増分バックアップを実行した方がよいと考えられます。
 - `--incremental` オプションと `--incremental-with-redo-log-only` オプションの両方を使用して、増分バックアップの時間をベンチマークテストし、Redo ログバックアップ技法が従来の増分バックアップ方法より高速かつ少ないオーバーヘッドで実行するかどうかを確認します。結果は、データのサイズ、DML アクティビティの量、Redo ログファイルのサイズによって異なることがあります。現実的なデータボリュームと現実的なワークロードを実行するサーバーでテストを行ってください。たとえば、巨大な Redo ログファイルがあり、増分バックアップの過程でそれらを読み取ることは、従来の増分技法を使用して、InnoDB データファイルを読み取る場合と同じくらい時間がかかる可能性があります。逆に、データボリュームが大きい場合、すべてのデータファイルを読み取って、わずかな変更されたページを見つけることは、はるかに小さな Redo ログファイルを処理するよりも非効率的である可能性があります。

`--incremental` オプションと同様に、増分の観点は InnoDB テーブルにのみ適用されます。デフォルトで、すべての InnoDB 以外のファイルと `.frm` ファイルも増分バックアップに含まれます。増分バックアップで InnoDB 以外のデータを除外するには、`--only-innodb` オプションを使用します。

- `--incremental-base=mode:argument`

コマンド行形式	<code>--incremental-base=mode:argument</code>
型	文字列

このオプションを使用すると、`mysqlbackup` は増分バックアップの実行に必要な情報を `--start-lsn` オプションからではなく、バックアップディレクトリ内のメタデータから取得します。これにより、一連の増分バックアップの実行時に、絶え間なく変化し、予測不可能な LSN 値を指定する必要を避けられます。代わりに、出力構文の **モード** と **引数** の組み合わせによって、前回のバックアップディレクトリを見つける方法を指定します。代替方法:

- `dir:directory_path`

プリフィクス `dir:` の後にディレクトリパスを指定します。パス引数は、前回のバックアップからのデータが格納されているルートディレクトリを指します。最初の増分バックアップでは、フルバックアップを保持するディレクトリを指定し、2 回目の増分バックアップでは、最初の増分バックアップを保持するディレクトリを指定するようになります。

- `history:last_backup`

プリフィクス `history:` の後に、このモードで唯一の有効な引数 `last_backup` を指定します。これにより、`mysqlbackup` に、該当するインスタンスの `backup_history` テーブルに記録されている最後の成功したバックアップから `end_lsn` 値をクエリーさせます。

警告

前回のバックアップが `--no-connection` オプションによって取得されたフルバックアップであった場合、`history:` モードは使用しないでください。これは常にバックアップ履歴の記録をオフにし、`--incremental-base` オプションのこのモードを使用した後続の増分バックアップでエラーが発生する可能性があります。

- `--start-lsn=LSN`

コマンド行形式	<code>--start-lsn=LSN</code>
---------	------------------------------

型	数値
---	----

増分バックアップでは、前回のバックアップに含まれる最高の LSN 値を指定します。この値は、前回のバックアップ操作の出力から、または前回のバックアップ操作の `backup_history` テーブルの `end_lsn` カラムから取得できます。常に `--incremental` オプションと一緒に使用し、`--incremental-base` オプションを使用する場合に必要ではなく、増分バックアップに `--incremental-with-redo-log-only` メカニズムを使用する場合に推奨されません。

- `--incremental-backup-dir=PATH`

コマンド行形式	<code>--incremental-backup-dir=PATH</code>
型	ディレクトリ名

増分バックアップからのデータを格納する場所を指定します。これは、後続の増分バックアップに `--incremental-base` を使用した場合に、そのオプションで指定した場所と同じになります。

例 5.4 増分バックアップ

これらの例に、増分バックアップで使用される一般的なオプションの組み合わせを示します。

```
mysqlbackup --incremental \
--incremental-backup-dir=/var/mysql/backup/latest \
--incremental-base=dir:/var/mysql/backup/previous \
... backup

mysqlbackup --incremental-with-redo-log-only \
--incremental-backup-dir=/var/mysql/backup/latest \
--incremental-base=dir:/var/mysql/backup/previous \
... backup

mysqlbackup --incremental --start-lsn=12345 \
--incremental-backup-dir=/var/mysql/backup/inc \
... backup

mysqlbackup --incremental-with-redo-log-only --start-lsn=12345 \
--incremental-backup-dir=/var/mysql/backup/inc \
... backup
```

5.1.9 部分バックアップとリストアオプション

注記

MySQL Enterprise Backup 3.10 以降、2 つのオプション `--include-tables` と `--exclude-tables` が導入されました。これらは、新しいオプションと互換性がなく、将来のリリースで非推奨になる `--include`、`--databases`、`--databases-list-file`、および `--only-innodb-with-frm` の古いオプションを置き換えることを目的としています。参考のため、古いオプションに関する情報をこのセクションの最後の「レガシー部分バックアップオプション」に記載しています。

バックアップまたはリストアする特定のデータを選択するには、このセクションで説明する部分バックアップおよびリストアオプションを使用します。

部分バックアップの概要と部分バックアップの次のオプションに関する使用情報については、[セクション 3.3.4 「部分バックアップの作成」](#) を参照してください。

- `--include-tables=REGEXP`

コマンド行形式	<code>--include-tables=REGEXP</code>
型	文字列

完全修飾名 (`db_name.table_name` の形式) が正規表現 `REGEXP` に一致するテーブル (innodb と innodb 以外の両方) のみをバックアップやリストアに含めます。使用される正規表現構文は、POSIX 1003.2 規格に指定された拡張形式です。たとえば、`--include-tables='^mydb\.[t12]$'` はデータベース `mydb` 内のテーブル `t1` と `t2` に一致します。Unix のようなシステムでは、適宜、正規表現を引用符で囲んで、シェルメタキャラクタが解釈されないようにしてください。特殊文字 (スペース、ダッシュ、ピリオドなど) を含むデータベース名やファイル名を選択するためにこのオプションを使用する場合、いくつかの制限が適用されます。詳細については、[付録 A 「MySQL](#)

[Enterprise Backup の制限](#) のこの説明を参照してください。正規表現と一緒に指定せずに、オプションを使用すると、`mysqlbackup` はエラーをスローします。

このオプションはさまざまな種類のバックアップで使用できますが、選択的なリストアは、[トランスポートブルテーブルスペース \(TTS\)](#) を使用して作成されたバックアップ (つまり `--use-tts` オプションで作成されたバックアップ) でのみサポートされます。

このオプションはレガシー `--include`、`--databases`、`--databases-list-file`、または `--only-innodb-with-frm` オプションと一緒に使用することはできません。

`--exclude-tables` オプションと一緒に使用すると、`--include-tables` が最初に適用されます。つまり、`mysqlbackup` はまず `--include-tables` で指定されたすべてのテーブルを選択してから、`--exclude-tables` で指定されたテーブルをセットから除外します。

- `--exclude-tables=REGEXP`

コマンド行形式	<code>--exclude-tables=REGEXP</code>
型	文字列

完全修飾名 (`db_name.table_name` の形式) が正規表現 `REGEXP` に一致するすべてのテーブル (`innodb` と `innodb` 以外の両方) をバックアップやリストアから除外します。正規表現構文は、POSIX 1003.2 規格に指定された拡張形式です。たとえば、`--exclude-tables=^mydb\.t[12]$` はデータベース `mydb` 内のテーブル `t1` と `t2` に一致します。Unix のようなシステムでは、適宜、正規表現を引用符で囲んで、シェルメタキャラクタが解釈されないようにしてください。特殊文字 (スペース、タブ、ピリオドなど) を含むデータベース名やファイル名を選択するためにこのオプションを使用する場合、いくらかの制限が適用されます。詳細については、[付録A「MySQL Enterprise Backup の制限」](#) のこの説明を参照してください。正規表現と一緒に指定せずに、オプションを使用すると、`mysqlbackup` はエラーをスローします。

このオプションはさまざまな種類のバックアップで使用できますが、選択的なリストアは、[トランスポートブルテーブルスペース \(TTS\)](#) を使用して作成されたバックアップ (つまり `--use-tts` オプションで作成されたバックアップ) でのみサポートされます。

このオプションは `--include`、`--databases`、`--databases-list-file`、または `--only-innodb-with-frm` オプションと一緒に使用することはできません。

`--include-tables` オプションと一緒に使用すると、`--include-tables` が最初に適用されます。つまり、`mysqlbackup` はまず `--include-tables` で指定されたすべてのテーブルを選択してから、`--exclude-tables` で指定されたテーブルをバックアップのセットから除外します。

- `--only-known-file-types`

バックアップ専用。デフォルトで、データディレクトリ内のすべてのファイルがバックアップに含まれます (詳細は[セクション1.4「バックアップされるファイル」](#)を参照してください)。`--only-known-file-types` オプションを指定

した場合、`mysqlbackup` は、次の拡張子を持つ MySQL またはその組み込みのストレージエンジンのデータファイルであるファイルの種類のみをバックアップします。

- `.ARM`: ストレージエンジンメタデータをアーカイブします。
- `.ARZ`: ストレージエンジンデータをアーカイブします。
- `.CSM`: CSV ストレージエンジンデータ。
- `.CSV`: CSV ストレージエンジンデータ。
- `.frm`: テーブル定義。
- `.MRG`: ストレージエンジン参照をほかのテーブルにマージします。
- `.MYD`: MyISAM データ。
- `.MYI`: MyISAM インデックス。
- `.OPT`: データベース構成情報。
- `.PAR`: パーティション定義。
- `.TRG`: トリガーパラメータ。
- `.TRN`: トリガー名前空間情報。

• `--only-innodb`

バックアップ専用。InnoDB データとログファイルのみをバックアップします。ほかのストレージエンジンによって作成されたすべてのファイルが除外されます。一般に `mysqld` への接続が許可されない場合または MyISAM をコピーする必要がない場合に使用されます。

このオプションは `--slave-info` オプションと互換性がありません。

デフォルト: バックアップにはすべてのストレージエンジンからのファイルが含まれます。

• `--use-tts[={with-minimum-locking|with-full-locking}]`

コマンド行形式	<code>--use-tts[={with-minimum-locking with-full-locking}]</code>
型	列挙
デフォルト	<code>with-minimum-locking</code>
有効な値	<code>with-minimum-locking</code> <code>with-full-locking</code>

トランスポータブルテーブルスペース (TTS) を使用した InnoDB テーブルの選択的バックアップを有効にします。これは `--include-tables` オプションおよび `--exclude-tables` オプションと一緒に使用し、正規表現によってバックアップする InnoDB テーブルを選択します。バックアップで TTS を使用すると、次の利点があります。

- 別のサーバーにバックアップをリストアできます
- **システムテーブルスペース** がバックアップされないため、ディスク領域と I/O リソースを節約します
- テーブルのデータ整合性が MySQL Enterprise Backup によって管理されます

ただし、このオプションには次の制限があります。

- MySQL バージョン 5.6 以降のみをサポートします (MySQL の初期バージョンでは TTS をサポートしていないため)
- 独自の個別のテーブルスペースに格納されているテーブル (つまり、`innodb_file_per_table` オプションを有効にして作成されたテーブル) のみをバックアップできます

- パーティション化されたテーブルをバックアップできません
- 増分バックアップに使用できません

オプションに指定可能な値は 2 つあります。

- `with-minimum-locking`: 選択したテーブルのホットコピーがバックアップされ、[Redo ログ](#) (ホットバックアップ後に行われた関連の変更を含む部分のみによる) がバックアップに含まれる間、テーブルは読み取り専用モードでロックされます。ロックフェーズ中に作成されたすべてのテーブルは無視されます。
- `with-full-locking`: 選択したテーブルはバックアップ中に読み取り専用モードでロックされます。[Redo ログ](#) はバックアップに含まれません。ロックフェーズ中に作成されたすべてのテーブルは無視されます。

デフォルト: 最小のロックでバックアップ

`--use-fts` オプションを使用するには、`mysqlbackup` がサーバーに接続するためにユーザーの特別な権限が必要です。詳細については、[セクション 3.1.2 「MySQL の権限をバックアップ管理者に与える」](#) を参照してください。

`--use-fts` オプションを使用して作成されたバックアップをリストアするには、いくつかの特別な要件があります。詳細については「[Restoring Backups Created with the --use-fts Option](#)」を参照してください。

レガシー部分バックアップオプション

重要

このサブセクションの情報は、`--include`、`--databases`、`--databases-list-file`、および `--only-innodb-with-frm` のレガシーオプションを使用するためだけのもので、これらのオプションは次回の発行時に非推奨になります。部分バックアップの作成には、代わりに `--include-tables` と `--exclude-tables` の新しいオプションを使用することを強くお勧めします。部分バックアップのレガシーオプションと新しいオプションを単一のコマンドで組み合わせることはできません。

レガシーオプションに加えて、以下ではほかのオプションについても説明していますが、情報はレガシー部分バックアップオプションと一緒にオプションを使用する場合に限られます。

部分バックアップの概要とこれらのレガシーオプションに関する使用情報については、[Making a Partial Backup with the Legacy Options](#) を参照してください。

- `--include=REGEXP`

このオプションはバックアップする InnoDB テーブルのフィルタ用です。InnoDB テーブルの完全修飾名が、オプションで指定された正規表現に対してチェックされます。REGEXP が `db_name.table_name` に一致する場合、そのテーブルは含まれます。使用される正規表現構文は、POSIX 1003.2 規格に指定された拡張形式です。たとえば、`--include=mydb\.[12]` はデータベース `mydb` 内のテーブル `t1` と `t2` に一致します。正規表現と一緒に指定せずにこのオプションが使用された場合、`mysqlbackup` はエラーをスローします。

このオプションは、MySQL オプション `innodb_file_per_table` が有効にされて (これは MySQL 5.6 以降のデフォルトの設定です) 作成された InnoDB テーブルにのみ適用されます。その場合テーブルはバックアップに含めたり、除外したりできる個別のファイルになります。InnoDB システムのテーブルスペース内のすべてのテーブルは常にバックアップされます。

指定した正規表現に一致する InnoDB テーブル名がない場合、一致がないことを示すメッセージとともにエラーがスローされます。

デフォルト: すべての InnoDB テーブルをバックアップします。

注記

このオプションは InnoDB 以外のテーブルをフィルタしないため、`--databases` や `--databases-list-file` などのオプションを使用できます。

重要

このオプションは InnoDB テーブルに関連付けられた `.frm` ファイルをフィルタしません。つまり、オプションの値に関係なく、ほかのオプションによって除外されていない限り、すべての InnoDB テーブルのすべての `.frm` ファイルが常にバックアップされます。InnoDB テーブルのバックアップしない `.frm` ファイルは、データベースバックアップをリストアする前に削除しておいてください。詳細については、[Making a Partial Backup with the Legacy Options](#)を参照してください。

- `--databases=LIST`

バックアップする InnoDB 以外のテーブルを指定します。この引数では、次の形式のデータベースまたはテーブル名のスペース区切りのリストを指定します。

```
"db_name[table_name] db_name1[table_name1] ..."
```

指定した値に一致するデータベースやテーブルがない場合、InnoDB 以外のデータファイルはバックアップされません。詳細については、「[レガシーオプションによる部分バックアップの作成](#)」を参照してください。

デフォルトですべてのデータベースのすべての InnoDB 以外のテーブルがバックアップされます。

注記

このオプションは、これで指定するデータベースやテーブルの InnoDB データファイル (`.ibd` ファイル) に対するフィルタ効果はありません。InnoDB データファイルをフィルタするには、代わりに `--include` オプションを使用します。

- `--databases-list-file=PATH`

バックアップする InnoDB 以外のテーブルを一覧表示するファイルのパス名を指定します。このファイルには、改行やスペースで区切られたデータベースまたは完全修飾テーブル名のエントリが含まれます。エントリの形式は `--databases` オプションの場合と同じです。

```
db_name[table_name]
db_name1[table_name1]
...
```

空白は自動的に削除されないため、データベースまたはテーブル名の前後の空白を削除します。コメントを含めるには、行を `#` 文字で始めます。正規表現は使用できません。

指定したエントリに一致するデータベースやテーブルがない場合、InnoDB 以外のデータファイルはバックアップされません。

注記

このオプションは、これで指定するデータベースやテーブルの InnoDB データファイル (`.ibd` ファイル) に対するフィルタ効果はありません。InnoDB データファイルをフィルタするには、代わりに `--include` オプションを使用します。

- `--only-innodb-with-frm[={all|related}]`

InnoDB テーブルに関連付けられた InnoDB データ、ログファイル、および `.frm` ファイルのみをバックアップします。

- `--only-innodb-with-frm=all` はすべての InnoDB テーブルの `.frm` ファイルをバックアップに含めます。
- `--only-innodb-with-frm=related` は `--include` オプションと組み合わせて、部分バックアップに含まれるテーブルの `.frm` ファイルのみをコピーします。
- 引数を付けない `--only-innodb-with-frm` は `--only-innodb-with-frm=related` と同じです。

注記

増分バックアップの場合、変更された `.ibd` ファイルのみがバックアップされても、指定されたすべての InnoDB テーブルに関連付けられた `.frm` が含まれます。

このオプションにより、通常 `FLUSH TABLES WITH READ LOCK` ステートメントによってサーバーが読み取り専用状態になっている間に実行する、InnoDB `.frm` ファイルのバックアップステップをスクリプト化する必要がなくなります。バックアップ操作が真の**ホットバックアップ**になり、データベースの処理を妨げないように、`.frm` ファイルはサーバーを読み取り専用状態にしなくてもコピーされます。バックアップの進行中に、`ALTER TABLE` やその他の DDL ステートメントによって、InnoDB テーブルの `.frm` ファイルが変更されないことを確認する必要があります。`mysqlbackup` コマンドが、バックアップ操作中に関連の `.frm` ファイルの変更を検出した場合、エラーで停止します。バックアップ操作時の InnoDB テーブルへの DDL を禁止することが現実的でない場合は、代わりに `--only-innodb` オプションを使用し、サーバーのロック中に `.frm` ファイルをコピーする従来の方法を使用します。

ほかのストレージエンジンによって作成されたすべてのファイルが除外されます。一般に `mysqld` への接続が許可されない場合または MyISAM をコピーする必要がない場合、たとえばバックアップ中に DDL の変更がないことが確信できる場合に使用されます。手順と例については、[Making a Partial Backup with the Legacy Options](#)を参照してください。

このオプションは `--slave-info` オプションと互換性がありません。

デフォルト: バックアップにはすべてのストレージエンジンからのファイルが含まれます。

- `--use-fts[={with-minimum-locking|with-full-locking}]`

トランスポータブルテーブルスペース (TTS) を使用した InnoDB テーブルの選択的バックアップを有効にします。これは、正規表現によってバックアップする InnoDB テーブルを選択する `--include` オプションとともに使用します。バックアップで TTS を使用すると、次の利点があります。

- 別のサーバーにバックアップをリストアできます
- システムテーブルスペースがバックアップされないため、ディスク領域と I/O リソースを節約します
- テーブルのデータ整合性が MySQL Enterprise Backup によって管理されます

ただし、このオプションには次の制限があります。

- MySQL バージョン 5.6 以降のみをサポートします (MySQL の初期バージョンでは TTS をサポートしていないため)
- 独自の個別のテーブルスペースに格納されているテーブル (つまり、`innodb_file_per_table` オプションを有効にして作成されたテーブル) のみをバックアップできます
- パーティション化されたテーブルをバックアップできません
- バックアップから選択してテーブルをリストアすることはできません
- 増分バックアップに使用できません

オプションに指定可能な値は 2 つあります。

- `with-minimum-locking`: 選択したテーブルのホットコピーがバックアップされ、Redo ログ (ホットバックアップ後に行われた関連の変更を含む部分のみによる) がバックアップに含まれる間、テーブルは読み取り専用モードでロックされます。ロックフェーズ中に作成されたすべてのテーブルは無視されます。
- `with-full-locking`: 選択したテーブルはバックアップ中に読み取り専用モードでロックされます。Redo ログはバックアップに含まれません。ロックフェーズ中に作成されたすべてのテーブルは無視されます。

デフォルト: 最小のロックでバックアップ

`--use-fts` オプションを使用して作成されたバックアップをリストアするには、いくつかの特別な要件があります。詳細については、[セクション 4.2 「リストア操作の実行」](#) の説明を参照してください。

5.1.10 単一ファイルバックアップオプション

これらのオプションは単一ファイルバックアップに関連付けられています。それらは、単一イメージバックアップをバックまたはアンパックする `mysqlbackup` サブコマンドの `backup-to-image`、`image-to-backup-dir`、`backup-dir-to-image`、`list-image`、および `extract` と組み合わせて使用します。使用情報については、[セクション 3.3.5 「単一ファイルバックアップの作成」](#) を参照してください。

- `--backup-image=IMAGE`

コマンド行形式	<code>--backup-image=IMAGE</code>
型	ファイル名

単一ファイルバックアップに使用するファイルのパス名を指定します。デフォルトで、単一ファイルバックアップは、テープバックアップや `ssh` 関連のネットワークコマンドなどのその他のコマンドに直接パイプできるように、標準出力にストリーミングされます。

オプションでイメージ名の前に `file:` を付けて、ファイル I/O を示すことができます (デフォルト)。テープバックアップの場合、イメージ名の前に `sbt:` を付けます。テープバックアップの詳細については、[セクション 3.3.5.2 「テープへのバックアップ」](#) を参照してください。

- `--src-entry=PATH`

コマンド行形式	<code>--src-entry=PATH</code>
型	パス名

単一ファイルバックアップから抽出するファイルやディレクトリを識別します。このオプションは `extract` コマンドで使用します。引数がディレクトリである場合、そのすべてのファイルとサブディレクトリの内容が抽出されます。引数にパターンマッチング表現は使用できません。オプションで、`--dst-entry` オプションを指定して、その元のパス名と異なる場所にあるファイルやディレクトリを抽出することもできます。

例: `src-entry=meta/comments.txt` は 1 つのファイル `comments.txt` のみを抽出し、`src-entry=meta` は `meta` サブディレクトリのディレクトリツリー全体を抽出します。

デフォルト: すべてのエントリが抽出されます。

- `--dst-entry=PATH`

コマンド行形式	<code>--dst-entry=PATH</code>
型	パス名

単一のファイルやディレクトリをユーザー指定のパスに抽出するために、単一ファイルバックアップで使用します。このオプションを使用するには、`--src-entry` オプションを指定する必要があります。このオプションは、`-src-entry=PATH` オプションで指定されたエントリに対応する、バックアップイメージ内の選択したエントリの宛先パスを指定します。エントリは単一ファイルまたは単一ディレクトリを指すことができます。たとえば、バックアップイメージからコメントファイルを取得し、それを `/tmp/my-comments.txt` という名前で保存するには、次のようなコマンドを使用します。

```
mysqlbackup --src-entry=meta/comments.txt \
--dst-entry=/tmp/my-comments.txt \
--backup-image=/var/myimage.bki extract
```

同様に、単一ファイルバックアップ内の `meta` ディレクトリのすべての内容を `/data/my-meta` という名前で抽出するには、次のようなコマンドを使用します。

```
mysqlbackup --src-entry=meta \
--dst-entry=/data/my-meta \
--backup-image=/var/myimage.bki extract
```

指定するパスはワイルドカード拡張子や正規表現を使用しない単純なパス名です。

デフォルト: デフォルトで、元のパス名が使用され、ローカルファイルシステムにファイルが作成されます。

- `--sbt-database-name=NAME`

コマンド行形式	<code>--sbt-database-name=NAME</code>
型	文字列
デフォルト	MySQL

テープバックアップの場合、このオプションは、メディアとポリシーの選択についてのメディア管理ソフトウェア (MMS) へのヒントとして使用できます。この名前は MySQL データベース名とは関係ありません。それは MMS によって使用される用語です。使用の詳細については、[セクション3.3.5.2「テープへのバックアップ」](#)を参照してください。

- `--sbt-lib-path=PATH`

コマンド行形式	<code>--sbt-lib-path=PATH</code>
型	ファイル名

テープバックアップを管理するソフトウェアによって使用される SBT ライブラリのパス名。これを指定しない場合、オペレーティングシステム固有の検索方法を使用して、`libobk.so` (UNIX) または `orasbt.dll` (Windows) が見つけられます。使用の詳細については、[セクション3.3.5.2「テープへのバックアップ」](#)を参照してください。

- `--sbt-environment=VAR=value,...`

コマンド行形式	<code>--sbt-environment=VAR1=value1[,VAR2=value2[,...]] SBT API provider)</code>
型	文字列

`mysqlbackup` の各呼び出しの前後での環境変数の設定と設定解除の代替方法として、製品固有の環境変数を Oracle Secure Backup または別の SBT 互換バックアップ管理製品に渡します。

このオプションのパラメータは、Oracle Database の RMAN ツールに似た構文を使用した鍵と値のペアのカンマ区切りのリストです。たとえば、`--sbt-environment=VAR1=val1,VAR2=val2,VAR3=val3` です。

環境変数によって制御可能な機能を確認するには、バックアップ管理製品のドキュメントを参照してください。たとえば、Oracle Secure Backup 製品では、`OB_MEDIA_FAMILY`、`OB_DEVICE`、および `OB_RESOURCE_WAIT_TIME` などの環境変数を定義します。`--sbt-environment="OB_MEDIA_FAMILY=my_mf,OB_DEVICE=my_tape"` などのオプションを指定して、`mysqlbackup` でそれらの変数を設定できます。

引数文字列にコマンドシェルによって認識される空白や特殊文字が含まれる場合は、引数文字列全体を引用符で囲みます。等号やカンマをエスケープするには、`\` 文字を使用します。たとえば、`--sbt-environment="VAR1=multiple words,VAR2=<angle_brackets>,VAR3=2+2\=4"` となります。

- `--disable-manifest`

`meta` サブディレクトリ内に存在する `backup_create.xml` および `backup_content.xml` である、バックアップ操作のマニフェストファイルの生成を無効にします。

5.1.11 パフォーマンス/スケーラビリティ/容量オプション

これらのオプションは、ビジーまたは巨大なデータベースのバックアップオーバーヘッドを最小にしたり、リソースの問題の発生時に、プロセスの動作を指定したりするために、バックアッププロセスによって使用されるリソースを制限します。

- `--number-of-buffers=num_buffers`

コマンド行形式	<code>--number-of-buffers=NUMBER</code>
型	数値
デフォルト	14
最小値	1

マルチスレッドオプション時に使用する、各 16M バイトのサイズのバッファ数を指定します。

バックアップ、特に圧縮を使用する場合など、CPU を大量に消費する処理には大きな数を使用します。バックアップのリストアなど、ディスクを大量に使用する処理には小さい数を使用します。この値は、操作の種類に応じて、少なくとも読み取りスレッドまたは書き込みスレッドの数と同じにするべきです。

デフォルト: 現在 14。

圧縮または増分バックアップ操作の場合、ヘッダーを収容するために、バッファサイズは 16M バイトよりやや大きくなります。

単一ファイルの増分バックアップと単一ファイルの圧縮バックアップには、1 つの追加バッファが使用されません。

圧縮バックアップ、圧縮単一ファイルバックアップ、および非圧縮 `apply-log` 操作には、プロセススレッドごとに 1 つの追加バッファが必要です。

読み取り、書き込み、および処理スレッドの数を変更する場合、それらのほかのオプションによって指定されたスレッドの合計数よりやや大きくなるように、この値の変更を試してみることができます。RAID や非 RAID ストレージ

ジデバイスなどのさまざまなハードウェア構成でのこのオプションとほかのパフォーマンス関連オプションの値の推奨される組み合わせに関する追加のアドバイスについては、[セクション7.1「バックアップパフォーマンスの最適化」](#)および[セクション7.2「リストアパフォーマンスの最適化」](#)を参照してください。

- `--read-threads=num_threads`

コマンド行形式	<code>--read-threads=NUMBER</code>
型	数値
デフォルト	1
最小値	1
最大値	15

ディスクからデータを読み取るために使用するスレッド数を指定します。

デフォルト: 現在 1。このデフォルトは、`copy-back`、`extract`、および `backup` 操作の種類に適用されます。0 の値を指定した場合、警告なしで 1 に調整されます。最大は 15 です。負の値を指定した場合、警告なく 15 に調整されます。`apply-log` 操作の場合、このオプションの設定に関係なく、読み取りスレッド数は常に 1 になります。RAID や非 RAID ストレージデバイスなどのさまざまなハードウェアオプション構成の `--read-threads`、`--process-threads`、および `--write-threads` の値の推奨される組み合わせに関するアドバイスについては、[セクション7.1「バックアップパフォーマンスの最適化」](#)および[セクション7.2「リストアパフォーマンスの最適化」](#)を参照してください。

- `--process-threads=num_threads`

コマンド行形式	<code>--process-threads=NUMBER</code>
型	数値
デフォルト	6
最小値	1
最大値	15

バックアップファイルの圧縮または圧縮解除などのデータの処理に使用するスレッド数を指定します。

デフォルト: 現在 6。このデフォルトは、`extract` および `backup` の操作の種類に適用されます。オプション `--incremental-with-redo-log-only`、`apply-incremental-backup`、`copy-back`、または `backup-dir-to-image` のいずれかを使用した場合、無視されます。

0 の値を指定した場合、警告なしで 1 に調整されます。最大は 15 です。負の値を指定した場合、警告なく 15 に調整されます。`apply-log` 操作の場合、このオプションの設定に関係なく、プロセススレッド数は常に 1 になります。RAID や非 RAID ストレージデバイスなどのさまざまなハードウェアオプション構成の `--read-threads`、`--process-threads`、および `--write-threads` の値の推奨される組み合わせに関するアドバイスについては、[セクション7.1「バックアップパフォーマンスの最適化」](#)および[セクション7.2「リストアパフォーマンスの最適化」](#)を参照してください。

- `--write-threads=num_threads`

コマンド行形式	<code>--write-threads=NUMBER</code>
型	数値
デフォルト	1
最小値	1

最大値	15
-----	----

ディスクにデータを書き込むために使用するスレッド数を指定します。

デフォルト: 現在 1。このデフォルトは、`copy-back`、`extract`、および `backup` 操作の種類に適用されます。単一ファイルバックアップオプションの `list-image` または `validate` のいずれかを使用すると、無視されます。

0 の値を指定した場合、警告なしで 1 に調整されます。最大は 15 です。負の値を指定した場合、警告なく 15 に調整されます。`apply-log` 操作の場合、このオプションの設定に関係なく、書き込みスレッド数は常に 0 になります。RAID や非 RAID ストレージデバイスなどのさまざまなハードウェアオプション構成の `--read-threads`、`--process-threads`、および `--write-threads` の値の推奨される組み合わせに関するアドバイスについては、[セクション 7.1 「バックアップパフォーマンスの最適化」](#) および [セクション 7.2 「リストアパフォーマンスの最適化」](#) を参照してください。

- `--limit-memory=MB`

コマンド行形式	<code>--limit-memory=MB</code>
型	数値
デフォルト	100 for <code>apply-log</code> (without uncompression), 300 for other operations
最小値	0
最大値	999999
Unit	megabyte

`mysqlbackup` コマンドで使用可能なメガバイト単位での最大メモリーを指定します。以前は `apply-log` 操作にのみ適用されましたが、MySQL Enterprise Backup 3.8 以降では、すべての操作に適用されます。オプションの値に `mb` や `kb` などのサフィクスを含めないでください。

デフォルト: `--uncompress` と一緒に使用しない `apply-log` の場合 100、すべての操作の場合 300 (メガバイト単位)。

この操作によって指定されたメモリー制限は、マルチスレッド処理に使用可能な 16M バイトのバッファ一数も制限します。たとえば、300M バイトの制限では、バッファ一数の最大数は 18 です。`--read-threads`、`--process-threads`、`--write-threads`、`--number-of-buffers` の値を増やすために、追加のバッファ一が必要な場合、`--limit-memory` 値もそれに比例して増やします。

- `--sleep=MS`

コマンド行形式	<code>--sleep=MS</code>
型	数値
デフォルト	0
Unit	millisecond

InnoDB テーブルから特定の量のデータをコピーした後に、スリープするミリ秒数を指定します。データの各ブロックは 1024 InnoDB データページで、一般に合計 16M バイトになります。これは、データベースサーバーの CPU と I/O オーバーヘッドを制限します。

デフォルト: 0 (自発的スリープなし)。

- `--no-locking`

接続が使用できる場合でも、InnoDB 以外のファイルのバックアップ時にロックを無効にします。通常のデータベース処理への少ない影響で、InnoDB 以外のデータをコピーするために使用できます。それらのファイルのバックアップ中に、何らかの変更があった場合、InnoDB 以外のデータに不整合が発生する可能性があります。

- `--page-reread-time=MS`

コマンド行形式	<code>--page-reread-time=MS</code>
---------	------------------------------------

型	数値
デフォルト	100
Unit	millisecond

チェックサムテストに失敗したページを再読み取りするまで、`mysqlbackup` が待機するミリ秒単位の間隔。ビジーなサーバーは `mysqlbackup` がページを読み取る同じ瞬間にページを書き込むことがあります。50 マイクロ秒を意味する 0.05 など、浮動小数点数を指定できます。最善の分解能は 1 マイクロ秒ですが、一部のプラットフォームではこれより悪くなることがあります。デフォルトは 100 ミリ秒 (0.1 秒) です。

- `--page-reread-count=retry_limit`

コマンド行形式	<code>--page-reread-count=number</code>
型	数値
デフォルト	500

ページでチェックサムテストに失敗した場合の再読み取りの最大試行回数。ビジーなサーバーは `mysqlbackup` がページを読み取る同じ瞬間にページを書き込むことがあります。各試行間で `--page-reread-time` オプションに基づいた一時停止をして、同じページでこの多くのチェックサムテストが連続して失敗する場合、バックアップは失敗します。デフォルトは 500 です。

- `--on-disk-full={abort|abort_and_remove|warn}`

コマンド行形式	<code>--on-disk-full=option</code>
型	列挙
デフォルト	<code>abort</code>
有効な値	<code>abort</code> <code>warn</code> <code>abort_and_remove</code>

バックアッププロセスでディスク容量不足の状況が検出された場合の動作を指定します。このオプションはバックアップ操作専用です (`backup`、`backup-and-apply-log`、および `backup-to-image`)。

- `abort`: バックアップディレクトリを削除せずに、バックアップを中止します。ディスクはいったいなままです。
- `abort_and_remove`: バックアップを中止し、バックアップディレクトリを削除します。
- `warn`: 30 秒ごとに警告メッセージを書き込み、ディスク領域を使用できるようになるまでバックアップを再試行します。
デフォルト: `abort`。
- `--skip-unused-pages`

InnoDB テーブルのバックアップ時にテーブルスペース内の未使用のページをスキップします。このオプションは `backup` および `backup-to-image` 操作に適用できますが、増分バックアップには適用できません。このオプションは `backup-and-apply-log` 操作によって無視されます。

`--skip-unused-pages` オプションによって作成されたバックアップは、`copy-back-and-apply-log` を使用してリストアできないことに注意してください。

未使用ページは、多くの場合データの一括削除によって発生する空きページです。バックアップ時に未使用ページをスキップすることで、このオプションはバックアップサイズ、さらに操作に必要なディスク領域と I/O リソースを削減できます。ただし、バックアップへの後続の `apply-log` 操作は、操作中に未使用ページがテーブルに再度挿入されるため、完了まで時間がかかります。

- `--skip-binlog`

バックアップにバイナリログファイルを含めません。バイナリログファイルはデフォルトで、すべての種類のオンラインバックアップ (フル、増分、圧縮、部分、単一ファイルなど) に含まれます。詳細は [セクション1.4「バックアップされるファイル」](#) を参照してください。リソース、パフォーマンス、その他の問題が発生した場合に、このオプションを使用して、バイナリログのバックアップをスキップします。

注記

一部の既知の問題のため、オフラインバックアップを作成するときや、`--no-locking` オプションで作成された完全バックアップに基づいた増分バックアップを作成するときには、ユーザーは常に `--skip-binlog` オプションを使用する必要があります。詳細は、[付録 A「MySQL Enterprise Backup の制限」](#) を参照してください。

- `--skip-relaylog`

バックアップにリレーログファイルを含めません。リレーログファイルはデフォルトで、スレーブサーバーのすべての種類のオンラインバックアップ (フル、増分、圧縮、部分、単一ファイルなど) に含まれます。詳細は [セクション1.4「バックアップされるファイル」](#) を参照してください。リソース、パフォーマンス、その他の問題が発生した場合に、このオプションを使用して、リレーログのバックアップをスキップします。

注記

スレーブでバックアップが進行中に、ユーザーが `FLUSH LOGS` ステートメントを実行すると、バックアッププロセスは失敗します。バックアップ時に `FLUSH LOGS` ステートメントが実行されることが予想され、リレーログをバックアップに含める必要がない場合、`--skip-relaylog` オプションを使用します。

- `--log-bin-index[=PATH]`

コマンド行形式	<code>--log-bin-index=PATH</code>
型	ファイル名
デフォルト	<code>data_dir/host_name-bin.index</code>

MySQL 5.5 とすべてのオフラインバックアップの場合: バイナリログファイルをバックアップに含めるために、使用中のすべてのバイナリログファイルを一覧表示した、MySQL サーバー上のインデックスファイルの絶対パス (ファイル名と拡張子を含む) を指定します (下に示すデフォルトのパスと異なる場合)。

デフォルト: `data_dir/host_name-bin.index`。

- `--relay-log-index[=PATH]`

コマンド行形式	<code>--relay-log-index=PATH</code>
型	ファイル名
デフォルト	<code>data_dir/host_name-relay-bin.index</code>

スレーブサーバーのオフラインバックアップの場合のみ: リレーログファイルをバックアップに含めるために、使用中のすべてのリレーログファイルを一覧表示した、MySQL サーバー上のインデックスファイルの絶対パス (ファイル名と拡張子を含む) を指定します (下に示すデフォルトのパスと異なる場合)。

デフォルト: `data_dir/host_name-relay-bin.index`。

- `--master-info-file[=PATH]`

コマンド行形式	<code>--master-info-file=PATH</code>
型	ファイル名

デフォルト	<code>data_dir/master.info</code>
-------	-----------------------------------

スレーブサーバーのオフラインバックアップの場合のみ: 情報ファイルをバックアップに含めるために、スレーブがそのマスターに関する情報を記録する情報ファイルの絶対パス (ファイル名と拡張子を含む) を指定します (下に示すデフォルトのパスと異なる場合)。

デフォルト: `data_dir/master.info`。

- `--relaylog-info-file[=PATH]`

コマンド行形式	<code>--relaylog-info-file=PATH</code>
型	ファイル名
デフォルト	<code>data_dir/relay-log.info</code>

スレーブサーバーのオフラインバックアップの場合のみ: 情報ファイルをバックアップに含めるために、スレーブがリレーログに関する情報を記録する情報ファイルの絶対パス (ファイル名と拡張子を含む) を指定します (下に示すデフォルトのパスと異なる場合)。

デフォルト: `data_dir/relay-log.info`。

- `--optimistic-time[=DATE-TIME]`

コマンド行形式	<code>--optimistic-time=DATE-TIME</code>
型	文字列
デフォルト	<code>now</code>

「オプティミスティック時間」(変更されていないテーブルが「非アクティブテーブル」とされるまでの時間)として、オプションで指定された値でオプティミスティックバックアップを実行します。「非アクティブテーブル」はバックアッププロセス中に変更される可能性がないとみなされます。非アクティブテーブルは、バックアップのオプティミスティックフェーズでバックアップされ、ほかのすべてのテーブルは通常フェーズでバックアップされます。オプティミスティックバックアップの概念、使用事例、コマンドサンプルの詳細については、[セクション 3.3.6「オプティミスティックバックアップの作成」](#)を参照してください。

オプションの指定で受け付けられる形式は次のようになります。

- `now`: これは、すべてのテーブルをバックアッププロセスのオプティミスティックフェーズに含めます。これは値が指定されていない場合のオプションのデフォルト値です。
- `{Number}{Unit}`: 過去の特定の期間の時間として、オプティミスティック時間を示します。`{Unit}` は `years`、`months`、`hours`、`minutes` のいずれかになります。この形式のオプション文字列の例には次が含まれます。`5years`、`2days`、`13months`、`23hours`、および `35minutes`。
- 次のいずれかの形式の日時形式: `YYMMDD`、`YYYYMMDD`、`YYMMDDHHMMSS`、`YYYYMMDDHHMMSSYY-MM-DD`、`YYYY-MM-DD`、`YY-MM-DD`、または `HH.MM.SSYYYYMMDDTHHMMSS` (ISO 8601 規格に指定されているとおり)。

`optimistic-time` オプションと `optimistic-busy-tables` オプションの両方を使用し、オプティミスティックフェーズでバックアップされるテーブルの決定で競合した場合、`optimistic-busy-tables` が `optimistic-time` より優先されます。

- `--optimistic-busy-tables=REGEXP`

コマンド行形式	<code>--optimistic-busy-tables=REGEXP</code>
型	文字列

バックアッププロセス中に変更される可能性があるため、オプティミスティックバックアップの最初のフェーズでスキップされるテーブルを選択するオプションで指定した正規表現を使用して、オプティミスティックバックアップを実行します。完全修飾名 (`database_name.table_name` の形式) が正規表現によって一致するテーブルは、「ビジーテーブル」とされ、これはバックアップの第 2 または「通常」フェーズでバックアップされます。完全修飾名が正規表現によって一致しないテーブルは、「非アクティブテーブル」とされ、これはバックアップの第 1 また

は「オプティミスティック」フェーズでバックアップされます。オプティミスティックバックアップの概念、使用事例、コマンドサンプルの詳細については、[セクション3.3.6「オプティミスティックバックアップの作成」](#)を参照してください。

オプションが使用されていても、それとともに正規表現が指定されていない場合、MySQL Enterprise Backup はエラーをスローします。

`optimistic-time` オプションと `optimistic-busy-tables` オプションの両方を使用し、「オプティミスティック」にするテーブルの決定で競合した場合、`optimistic-busy-tables` が `optimistic-time` より優先されます。

5.1.12 メッセージロギングオプション

`mysqlbackup` は重要な進行状況とエラー情報を `stderr` ストリームに書き込みます。この情報は、多くの場合に操作中に発生する問題を突き止めるためにきわめて貴重です。MySQL Enterprise Backup 3.9 から、任意のデバッグプロセスでエラー情報に簡単にアクセスできるように、`stderr` ストリームへの出力はデフォルトでログファイルにも保存されます (ほとんどの `mysqlbackup` 操作で)。

メッセージロギングは Unix のようなシステムの `tee` プロセスのように機能し、プログラムの出力が分離されて、表示され、ファイルに保存されます。生成されるログファイルは次の形式で名前が付けられます。 `MEB_timestamp_operation.log`。ここで `operation` は実行された `mysqlbackup` 操作 (`backup`、`apply-log` など) で、`timestamp` は操作が実行された日時です。これはログファイルの名前の例です。

```
MEB_2013-06-24.16-32-43_backup.log
MEB_2013-06-28.11-07-18_apply_log.log
MEB_2013-06-29.10-08-06_list_image.log
```

次のオプションはメッセージロギング機能を制御します。

- `--skip-messages-logdir`

メッセージロギングをスキップします。ロギングはデフォルトでオンにされ (`list-image` および `validate` 操作を除きます。詳細については、`--messages-logdir` オプションの説明を参照してください)、このオプションによってオフにされます。

- `--messages-logdir=path`

コマンド行形式	<code>--messages-logdir=PATH</code>
型	ディレクトリ名
デフォルト	<code>backup_dir/meta</code>

メッセージログを格納するための既存のディレクトリのパス名を指定します。指定したディレクトリが存在しない場合、メッセージロギングが失敗し、エラーメッセージを返します。このオプションを省略した場合、`backup_dir/meta` のデフォルトのディレクトリが使用されます。

注記

このオプションを使用して、`list-image` および `validate` 操作のメッセージロギングをオンにします。2つの操作では、ファイルが変更されず、通常それらのデバッグにメッセージログが必要ないため、メッセージロギングはデフォルトでオフにされます。さらに、2つの操作では `backup_dir/meta` のデフォルトのパス名は意味がないため、メッセージロギングをオンにし、ログファイルを保存するディレクトリのパス名を指定するために、このオプションが必要です。ただし、`--skip-messages-logdir` オプションも指定した場合、それが優先され、メッセージロギングがスキップされます。

次は、メッセージロギングの制御方法を示す例です。

これによって、デフォルトの設定のため、ディレクトリ `/home/backup_dir/meta` に `backup` 操作のログファイルが作成されます。

```
$MYSQLBACKUP -uroot --port=3306 --backup-dir=/home/backup_dir backup
```

これは、`backup` 操作のメッセージロギングをスキップします。

```
$MYSQLBACKUP -uroot --port=3306 --backup-dir=/home/backup_dir \
--skip-messages-logdir backup
```

これにより、`apply-log` 操作のログファイルが、デフォルトの場所ではなく、`/home/teelog_dir` という既存のディレクトリに作成されます。

```
$MYSQLBACKUP -uroot --port=3306 --backup-dir=/home/backup_dir \
--messages-logdir=/home/teelog_dir apply-log
```

これにより、`list-image` 操作のログファイルが、`/home/teelog_dir` という既存のディレクトリに作成されます。

```
$MYSQLBACKUP -uroot --port=3306 --backup-image=/backup/my.mbi \
--messages-logdir=/home/teelog_dir list-image
```

5.1.13 進行状況レポートオプション

`mysqlbackup` の進行状況レポート機能を制御するために、`--show-progress` と `--progress-interval` の 2 つのオプションがあります。

- `--show-progress[={stderr|stdout|file:FILENAME|fifo:FIFONAME|table|variable}]`

コマンド行形式	<code>--show-progress[=destinations]</code>
型	列挙
有効な値	stderr stdout file:FILENAME fifo:FIFONAME table variable

このオプションは `mysqlbackup` に、その操作に関して、進行状況インジケータと呼ばれる短い進行状況レポートを定期的に出力するように指示します。

オプションの引数は、進行状況インジケータの送信先を制御します。

- `stderr`: 進行状況インジケータは標準エラー streams に送信されます。レポートはタイムスタンプ付きの `mysqlbackup INFO` メッセージに埋め込まれます。例:

```
130607 12:22:38 mysqlbackup: INFO: Progress: 191 of 191 MB; state: Completed
```

- `stdout`: 進行状況インジケータは標準出力 streams に送信されます。各進行状況インジケータの後に、単一の改行文字が出力されます。
- `file:FILENAME`: 進行状況インジケータがファイルに送信されます。新しい各進行状況レポートによってファイルが上書きされ、ファイルに最新の進行状況インジケータとそのあとに単一の改行文字が続いて含まれます。
- `fifo:FIFONAME`: 進行状況インジケータはファイルシステム FIFO に送信されます。各進行状況インジケータの後に、単一の改行文字が出力されます。

警告

FIFO を読み取るプロセスがない場合、`mysqlbackup` プロセスは実行の終了時に停止します。

- `table`: 進行状況インジケータは `mysql.backup_progress` テーブルに送信されます。これには MySQL サーバーへの接続が必要であるため、実行中の MySQL インスタンスをバックアップする場合にのみ機能します。`mysqlbackup` はまず進行状況レポートの 1 行を `mysql.backup_progress` テーブルに追加し、次に最新の進行状況インジケータで以降の行を更新します。進行状況インジケータはテーブルの `current_status` カラムに格納されます。

バックアップによって、MySQL インスタンスがロックされた場合 (たとえば、[FLUSH TABLES WITH READ LOCK](#) ステートメントの発行によって)、MySQL インスタンスのロックが解除されるまで、進行状況レポートが `mysql.backup_progress` テーブルに配信されません。

- `variable`: 進行状況インジケータはシステム変数 `backup_progress` に送信されます。

警告

MySQL Server にシステム変数 `backup_progress` がまだ定義されていません。ユーザーは変数を定義するために独自のプラグインを作成する必要があります。ユーザープラグインの詳細については、[MySQL プラグイン API](#)を参照してください。

`--show-progress` に引数が指定されていない場合、進行状況インジケータは `stderr` に送信されます。

コマンド行に `--show-progress` オプションを複数回指定して、進行状況を複数の宛先にレポートすることができます。たとえば、次のコマンド行はバックアップコマンドの進行状況を `stderr` と `meb_output` と呼ばれるファイルにレポートします。

```
mysqlbackup --show-progress --show-progress=file:meb_output --backup-dir=/full-backup
```

```
backup
```

進行状況インジケータは、`mysqlbackup` 操作の実行の進行状況を示す短い文字列です。進行状況インジケータは、操作の進行状況を測定する 1 つ以上のメーターから構成されます。例:

```
Progress: 100 of 1450 MB; state: Copying .ibd files
```

これは、これまで合計 1450M バイトのうち 100M バイトがコピーまたは処理され、`mysqlbackup` が現在 InnoDB データファイル (`.ibd` ファイル) をコピーしていることを示しています。

進行状況インジケータ文字列は `Progress:` から始まり、後に進行状況を測定する 1 つ以上のメーターが続きます。複数のメーターが存在する場合、それらはセミコロンで区切られます。さまざまな種類のメーターには次のものが含まれます。

- 合計データメーター: これは常に進行状況インジケータの最初のメーターになります。次の形式になります。

```
DATA of TOTAL UNIT
```

`DATA` と `TOTAL` は符号なしの 10 進整数で、`UNIT` は MB (メガバイト)、KB (キロバイト)、またはバイト (1MB=1024KB および 1KB=1024 バイト) です。

合計データメーターは、`mysqlbackup` 操作に応じて 2 つのやや異なる意味を持ちます。

- `mysqlbackup` 操作によって、コピーまたは処理されるデータの量とコピーまたは処理されるデータの合計量。例:

```
Progress: 200 of 1450 MB
```

操作がたとえば `backup` の場合、インジケータは 1450M バイトのうち 200M バイトがコピーされたことを意味します。ただし、操作がたとえば `validate` や `incremental` の場合、1450M バイトのうち 200M バイトが処理されたことを意味します。

- コピーまたは処理されたデータの合計量と操作の終了までにコピーされる合計の推定。推定合計量は、コマンドの実行の進行に伴い、サーバー上のデータに従って更新されます。

`backup` などの一部の操作では、実行の開始時にコピーまたは処理されるデータの量を正確に知ることはできません。そのため、合計データメーターには、バックアップされる推定合計データ量が示されます。推定量はコマンドの実行中に更新されます。例:

```
Progress: 200 of 1450 MB
```

その後次のようになります。

```
Progress: 200 of 1550 MB
```

サーバーに 100M バイトのデータが追加された場合。

操作が成功すると、最終の進行状況インジケータに、操作の終了時にコピーされたデータの実際の量が示されます。

- 圧縮メーター: $(\text{orig_size} - \text{compressed_size}) / \text{orig_size}$ として圧縮されたデータの各ブロックに定義される圧縮率の変動平均を示します。例:

```
compression: 40%
```

これは圧縮後、データが占める領域が 40% 縮小されることを意味します (最後の 10 データブロックの平均として計算)。

`mysqlbackup` 操作で `--compress` オプションを有効にした場合、圧縮メーターが進行状況インジケータに含まれます。圧縮メーターの値は、10 データブロック以上が圧縮されるまで定義されません。未定義のメーター値はメーターに「-」で示されます。

```
compression: -
```

- 状態メーター: コマンドが現在実行している主要なステップの短い説明です。例:

state: Copying InnoDB data

state: Waiting for locks

state: Copying system tablespace

state: Copying .ibd files

state: Copying non-InnoDB data

state: Completed

これは、さまざまなメーターによる進行状況インジケータの例です。

Progress: 300 of 1540 MB; state: Waiting for locks

Progress: 400 of 1450 MB; state: Copying InnoDB data: compression: 30%

進行状況インジケータに含まれる正確なメーターのセットは、それに対して使用されるコマンドとオプションによって異なります。

- `--progress-interval=SECONDS`

コマンド行形式	<code>--progress-interval=SECONDS</code>
型	数値
デフォルト	2
最小値	1
最大値	100000
Unit	second

進行状況レポートの秒単位での間隔。デフォルト値は 2 秒です。最短の間隔は 1 秒で、最大許容間隔は 100000 秒です。

5.1.14 暗号化オプション

これらのオプションは、暗号化された単一ファイルバックアップの作成とそれらの復号化用です。MySQL Enterprise Backup の暗号化および復号化機能の詳細と使用情報については、第8章「バックアップの暗号化」を参照してください。

- `--encrypt`

`backup-to-image` 操作によってバックアップイメージを作成する場合、または `backup-dir-to-image` サブコマンドでバックアップディレクトリを単一ファイルにパックする場合に、データを暗号化します。`backup` または `backup-and-apply-log` サブコマンドでは使用できません。

- `--decrypt`

`extract`、`image-to-backup-dir`、または `copy-back-and-apply-log` 操作の実行時に、暗号化されたバックアップイメージを復号化します。暗号化されたバックアップイメージへの `validate` または `list-image` 操作を実行する場合にも使用されます。

このオプションは、`apply-log`、`backup-and-apply-log`、または `copy-back` 操作では使用できません。`copy-back` サブコマンドを使用したリストアの場合、`image-to-backup-dir` または `extract` サブコマンドと `--decrypt` オプションを使用して、まず暗号化されたバックアップイメージをアンパックし、復号化する必要があります。

- `--key=STRING`

コマンド行形式	<code>--key=KEY</code>
型	文字列

バックアップイメージの暗号化および復号化のための対称鍵。64 文字の 16 進文字列としてエンコードされた 256 ビット鍵にしてください。鍵の作成方法については、[第8章「バックアップの暗号化」](#)を参照してください。このオプションは `--key-file` オプションと互換性がありません。

- `--key-file=PATH`

コマンド行形式	<code>--key-file=FILE</code>
型	ファイル名

バックアップイメージの暗号化および復号化のための 64 文字の 16 進文字列としてエンコードされた 256 ビット鍵を格納するファイルのパス名。このオプションは `--key` オプションと互換性がありません。

5.1.15 クラウドストレージオプション

これらのオプションは、クラウドストレージへの単一ファイルバックアップの作成またはリストア用です。MySQL Enterprise Backup でクラウドストレージを使用する場合の詳細と手順については、[セクション3.3.5.3「クラウドストレージへのバックアップ」](#)を参照してください。

- `--cloud-service`

データのバックアップまたはリストア用のクラウドサービス。現在 Amazon S3 サービスのみがサポートされており、`mysqlbackup` がこのオプションで受け付ける値は「s3」のみです。

- `--cloud-bucket`

バックアップイメージ用の Amazon S3 のストレージバケット。このオプションは、クラウドバックアップに Amazon S3 を使用する場合にのみ意味があります。

- `--cloud-object-key`

バックアップイメージ用の Amazon S3 オブジェクト鍵。このオプションは、クラウドバックアップに Amazon S3 を使用する場合にのみ意味があります。

- `--cloud-access-key-id`

Amazon S3 にログオンするための AWS アクセス鍵 ID。このオプションは、クラウドバックアップに Amazon S3 を使用する場合にのみ意味があります。

- `--cloud-secret-access-key`

`--cloud-access-key-id` に使用される AWS アクセス鍵 ID が付けられた AWS 秘密アクセス鍵。このオプションは、クラウドバックアップに Amazon S3 を使用する場合にのみ意味があります。

- `--cloud-aws-region`

`mysqlbackup` が S3 にアクセスする Amazon Web Services のリージョン。このオプションは、クラウドバックアップに Amazon S3 を使用する場合にのみ意味があります。

- `--cloud-trace`

クラウド操作のトレース情報を出力します。これは、`mysqlbackup` の非クラウド操作のトレースレベルを指定する `--trace` とは関係なく機能します。

オプションの 0 以外の値で、トレース機能が有効になります。デフォルト値は「0」です。

- `--cloud-proxy=proxy-url:port`

Amazon S3 にアクセスするために、環境のデフォルトのプロキシ設定を上書きするプロキシアドレスとポート番号。

注記

`list-image` 操作は、クラウドプロキシで Range ヘッダーをサポートしている場合にのみ、クラウドバックアップに対して実行できます。

5.1.16 特別なバックアップの種類オプション

これらのオプションは、レプリケーションで特定の役割を持つか、またはバックアップに特別な注意を必要とする特定の種類のデータを格納するデータベースサーバーのバックアップ用です。

- `--slave-info`

レプリケーションスレーブサーバーのバックアップ時、このオプションは同一のスレーブサーバーをセットアップするために必要な情報をキャプチャーします。これは、マスターサーバーのバイナリログの位置とバイナリログファイルの名前の `CHANGE MASTER` ステートメントを含むファイル `meta/ibbackup_slave_info` をバックアップディレクトリ内に作成します。この情報は `mysqlbackup` の出力にも出力されます。このマスターの新しいスレーブをセットアップするには、バックアップデータを別のサーバーにリストアし、バックアップデータでスレーブサーバーを起動し、`ibbackup_slave_info` ファイルに保存されたバイナリログ位置で、`CHANGE MASTER` コマンドを発行します。手順については、[セクション6.1「新しいレプリケーションスレーブのセットアップ」](#)を参照してください。

注記

このオプションはスレーブサーバーをバックアップする場合にのみ使用します。その動作は、マスターまたはレプリケーションサーバー以外で使用した場合は不定です。

このオプションは `--no-locking` オプションと互換性がありません。両方のオプションを一緒に使用すると、`mysqlbackup` はエラーをスローします。

このオプションは `--only-innodb` または `--only-innodb-with-frm` オプションと互換性がありません。

- `--suspend-at-end`

このオプションは、バックアップ手順が終わりに近づいたときに、`mysqlbackup` コマンドを一時停止します。これは、バックアップロググループホームディレクトリに `ibbackup_suspended` と呼ばれるファイルを作成し、ユーザーがそのファイルを削除するまで待機してから続行します。このオプションは、カスタムスクリプトによって、ロック動作と InnoDB 以外のファイルのバックアップをカスタマイズするために役立ちます。

中断する前にすべてのテーブルがロックされ、`--no-locking` または `--no-connection` オプションによってロックをオフにしない限り、データベースが読み取り専用状態になります。`--only-innodb` および `--only-innodb-with-frm` オプションはロックステップも妨げます。すべてのテーブルをロックすることは、ビジーなサーバーで問題が発生する可能性があるため、`--only-innodb` と `--suspend-at-end` の組み合わせを使用して、特定の InnoDB 以外のテーブルのみをバックアップすることができます。

- `--exec-when-locked="utility arg1 arg2 ..."`

コマンド行形式	<code>--exec-when-locked="utility arg1 arg2 ..."</code>
型	文字列

このオプションを使用して、通常のバックアップに含まれない情報をバックアップするスクリプトを書くことができます。たとえば `mysqldump` を使用して、MEMORY ストレージエンジンからディスク上にないテーブルをバックアップすることができます。

スクリプト内で、`BACKUP_DIR` 環境変数を設定して、現在のバックアップディレクトリを指します。次の例のように、引用符を使用して、`$BACKUP_DIR` の早期の拡大を防ぎます。

Unix または Linux システムの場合:

```
mysqlbackup --exec-when-locked='mysqldump mydb t1 > $BACKUP_DIR/t1.sql' other_options
```

Windows システムの場合:

```
mysqlbackup --exec-when-locked="mysqldump mydb t1 > %BACKUP_DIR%/t1.sql" other_options
```

ユーティリティーを実行できないか、ゼロ以外の終了ステータスを返す場合、バックアッププロセス全体がキャンセルされます。`--suspend-at-end` オプションも使用する場合、一時停止後に `--exec-when-locked` によって指定されたユーティリティーが実行されます。

5.2 構成ファイルとパラメータ

`mysqlbackup` オプションは、コマンド行または構成ファイル内の構成パラメータとして指定できます。このセクションでは、構成ファイルの使用について説明します。

一般に、`mysqlbackup` は構成オプションの処理の `mysql` スタイルに従います。「`mysqlbackup`」および「`client`」グループオプションはコマンド行オプションとして渡します。指定したコマンド行オプションによって、構成ファイルの値がオーバーライドされ、オプションが重複している場合は、最後のインスタンスが優先されます。`mysqlbackup` は、`mysqld` への接続を使用できない場合、「`mysqld`」グループ内のオプションも読み取り、ソースリポジトリに関連するパラメータを検出します。

パラメータ名の下線文字は、ダッシュに置換でき、この同じ規則を使用する `mysqld` パラメータと同様に、シノニムとして処理されます。(詳細については、MySQL リファレンスマニュアルの「[コマンド行でのオプションの使用](#)」を参照してください。)このドキュメントでは一般に `SHOW VARIABLES` ステートメントの出力に一致するように、下線を使用して名前を表示しています。

オプションファイル

`mysqlbackup` コマンドはバックアップ元の MySQL データの場所を読み取ります (優先順で)。

- 可能な場合、実行中のデータベースからの接続情報。そのため、ほとんどの場合に、コマンド行または構成ファイルにほとんどのオプションを指定することを避けられます。
- `mysqlbackup` コマンド行に指定するパラメータ。このようにして、個々のバックアップジョブに特定のオプションを指定できます。
- MySQL 構成ファイル (デフォルトで Unix 上の `my.cnf` および Windows 上の `my.ini`)。パラメータは最初「`mysqlbackup`」グループ、次に「`client`」グループ内で検索されます。ほとんどのバックアップジョブに適用する一般的なパラメータを構成ファイルに入れることができます。

`mysqlbackup` は、初期バックアップステップではファイルを上書きしないため、バックアップディレクトリには古いバックアップファイルが格納されてはなりません。`mysqlbackup` は、すでに存在するファイルを作成するように求められると停止し、既存のバックアップの損傷を回避します。便宜上、`--with-timestamp` オプションを指定します。これは常にメインバックアップディレクトリに下に、バックアップジョブごとに一意のタイムスタンプされたサブディレクトリを作成します。

バックアップデータと一緒に保存される構成ファイル

バックアップデータの各セットには、構成パラメータの最小セットを含む構成ファイル `backup-my.cnf` が含まれます。`mysqlbackup` コマンドは、このバックアップデータに適用される設定を記録するためにこのファイルを生成します。`apply-log` プロセスなどの後続の操作では、このファイルからオプションを読み取り、バックアップデータの構成を判断します。

例 5.5 `backup-my.cnf` ファイルの例

これは、`mysqlbackup` によって生成された `backup-my.cnf` ファイルの例です。

```
[mysqld]
innodb_data_file_path=ibdata1:256M;ibdata2:256M:autoextend
innodb_log_file_size=256M
innodb_log_files_in_group=3
```

生成された `backup-my.cnf` ファイル内のすべてのパスは、単一のバックアップディレクトリを指します。検証とメンテナンスを簡単にするため、一般にバックアップするすべてのデータを、異なるディレクトリに分散させずに、単一のディレクトリ内に格納します。

バックアップ時に、後の段階 (リストア操作など) で必要な構成パラメータはバックアップディレクトリに生成される `backup-my.cnf` ファイルに記録されます。`backup-my.cnf` には最小限必要なパラメータのみが格納されるため、そのファイルを大きく変更することなく、バックアップを別の場所にリストアできます。たとえば、`innodb_data_home_dir` オプションと `innodb_log_group_home_dir` オプションは `backup-my.cnf` に入れることができますが、それらの値が `backup-dir` 値と同じ場合、それらは省略されます。

第 6 章 レプリケーションで MySQL Enterprise Backup を使用する

目次

6.1 新しいレプリケーションスレーブのセットアップ	97
6.2 スレーブデータベースのバックアップとリストア	98
6.3 マスターデータベースのリストア	98

MySQL レプリケーションを使用して、マスターサーバーと一連のスレーブサーバー間でデータを同期させるシステムでは、バックアップ操作とリストア操作が特に重要です。レプリケーション構成で、MySQL Enterprise Backup は、システムイメージ全体を処理し、新しいスレーブサーバーをセットアップしたり、スレーブサーバーへの不要な作業を避ける効率的な方法で、マスターサーバーをリストアしたりするために役立ちます。選択する複数のスレーブサーバーを設置すると、バックアップを実行する場所に関する柔軟性が高まります。バイナリログを有効にすると、最後のバックアップより後の時点でも、任意の時点へのリストアに関する柔軟性が高まります。

MySQL Server 5.6 以上での GTID のサポート

MySQL Enterprise Backup は MySQL 5.6 の **GTID 機能** をサポートします。

- GTID 機能は、`mysqlbackup` コマンドがジョブの進行状況と履歴を記録するために使用する CSV テーブルと互換性があります。
- GTID 機能を使用するサーバーをバックアップすると、`mysqlbackup` は、バックアップデータの一部として、ファイル `gtid_executed.sql` を生成します。このファイルには `GTID_PURGED` 構成オプションを設定する SQL ステートメントが含まれます。バックアップデータをスレーブサーバーにリストアした後に、`mysql` コマンドラインを使用して、このファイルを実行します。オプションで、`mysql` によってこれを実行する前に、このファイル内の `CHANGE MASTER` コマンドのコメントを解除して、必要な認証パラメータを追加することができます。
- GTID を使用していないサーバーの場合、前のように、`--slave-info` オプションを使用して、後で `ibbackup_slave_info` ファイルを編集して実行することができます。

6.1 新しいレプリケーションスレーブのセットアップ

MySQL レプリケーションを使用する場合、MySQL Enterprise Backup により、マスターをバックアップし、そのバックアップを新しいスレーブサーバーにリストアすることによって、マスターを停止することなく、スレーブデータベースをセットアップできます。

1. バックアップを取得して、それをスレーブサーバーに転送し、`apply-log` オプションを付けて `mysqlbackup` を使用してそれを準備し、新しいスレーブの適切なディレクトリに、リストアされたバックアップとログファイルを配置します。
2. 新しいスレーブの `my.cnf` ファイルを編集し、「`mysqld`」セクションの下に、`skip-slave-start` および `event_scheduler=off` を挿入します。
3. 新しいスレーブ `mysqld` (バージョン `>= 5.1`) を起動します。これはバックアップが認識する最新の MySQL バイナリログの位置を出力します。

```
...
InnoDB: Last MySQL binlog file position 0 128760128, file name ./hundo-bin.006
...
```

InnoDB は、トランザクションのコミット時に、そのテーブルスペースにバイナリログの位置情報のみを格納することに注意してください。現在のバイナリロギングの位置を InnoDB に認識させるには、バイナリロギングが有効になっている間に、少なくとも 1 つのトランザクションを実行する必要があります。

4. スレーブで `CHANGE MASTER` SQL コマンドを使用して、それを正しく初期化します。例:

```
CHANGE MASTER TO
MASTER_LOG_FILE='hundin-bin.006',
MASTER_LOG_POS=128760128;
```

5. マスターからコピーされたイベントのステータスを `SLAVESIDE_DISABLED` に設定します。例:

```
mysql> UPDATE mysql.event SET status = 'SLAVESIDE_DISABLED';
```

6. ステップ 2 でスレーブの `my.cnf` ファイルに追加した行 `skip-slave-start` および `event_scheduler=off` エントリを削除します。
7. スレーブサーバーを再起動します。レプリケーションが開始します。

6.2 スレーブデータベースのバックアップとリストア

スレーブデータベースをバックアップするには、`--slave-info` オプションをバックアップコマンドに追加します。

スレーブサーバーにバックアップをリストアするには、[セクション6.1「新しいレプリケーションスレーブのセットアップ」](#)に説明する同じ手順に従ってください。

警告

ステートメントベースレプリケーション (SBR) 設定 (詳細は「[レプリケーション形式](#)」を参照してください) で、スレーブの SQL スレッドが停止するたびに (たとえば、スレーブサーバーにバックアップをリストアするときに必ず発生するスレーブのシャットダウン時など)、開いているすべての一時テーブルがスレーブにレプリケートされません (詳細は「[レプリケーションと一時テーブル](#)」を参照してください)。つまり、一時テーブルがマスター上に作成される可能性がある場合、その一時テーブルがバックアップから失われ、不整合になることがあるため、`mysqlbackup` によって、SBR 設定でスレーブサーバーをバックアップすることは必ずしも安全ではないことを意味します。そのため、マスター上に一時テーブルが作成されないことがわかっている場合にのみ、SBR スレーブで `mysqlbackup` を使用してください。

6.3 マスターデータベースのリストア

レプリケーションマスターデータベースの破損の問題を修正するには、不要な SQL 操作がスレーブサーバーに伝播されないように注意して、バックアップをリストアできます。

1. マスターデータベースのバックアップを使用して、`apply-log` 操作を実行し、データベースをシャットダウンして、`copy-back` 操作を実行します。
2. スレーブでマスターのリカバリに必要なバイナリログを 2 回受け取ることにないように、マスター `my.cnf` ファイルを編集し、`log-bin` のコメントを解除します。
3. バイナリログをマスターにパイプしている間、一時的にスレーブのレプリケーションが停止する必要があります。スレーブで、次を実行します。

```
mysql> STOP SLAVE;
```

4. リストアされたバックアップで、マスター `mysqld` を起動します。

```
$ mysqld
...
InnoDB: Doing recovery: scanned up to log sequence number 0 64300044
InnoDB: Last MySQL binlog file position 0 5585832, file name
./omnibook-bin.002
...
```

InnoDB はバイナリログファイル (この場合 `./omnibook-bin.002`) とリカバリできた位置 (この場合 `5585832`) を出力します。

5. バイナリログファイルの残りをリストアされたバックアップにパイプします。たとえば、`omnibook-bin.002` の後に生成された 2 つの追加のバイナリログファイル `omnibook-bin.003` および `omnibook-bin.004` がある場合、サー

バーへの単一の接続で、それらすべてをパイプします (`mysqlbinlog` の使用の詳細については、「[バイナリログを使用したポイントインタイム \(増分\) リカバリ](#)」を参照してください)。

```
$ mysqlbinlog --start-position=5585833 /mysqldatadir/omnibook-bin.002 \  
/mysqldatadir/omnibook-bin.003 /mysqldatadir/omnibook-bin.004 | mysql
```

残りのバイナリログファイルの数は、最後のバックアップからデータベースを最新にしたいと考える時間までの期間の長さによって異なります。期間が長いほど、多くの残りのバイナリログファイルが存在します。リストアの成功には、その期間のすべての連続したバイナリログの位置を含むすべてのバイナリログファイルが必要です。

- これでマスターデータベースがリカバリされます。マスターをシャットダウンし、`my.cnf` を編集して、`log-bin` のコメントを解除します。
- 再度マスターを起動します。
- スレーブで再度レプリケーションを起動します。

```
mysql> START SLAVE;
```


第 7 章 MySQL Enterprise Backup のパフォーマンスに関する考慮事項

目次

7.1 バックアップパフォーマンスの最適化	101
7.2 リストアパフォーマンスの最適化	104

7.1 バックアップパフォーマンスの最適化

このセクションでは、MySQL Enterprise Backup 製品によるバックアップ操作のパフォーマンスの考慮事項について説明します。バックアップ手順を最適化し、チューニングする場合、raw パフォーマンス (バックアップの完了にかかる時間) とデータベースサーバー上のオーバーヘッドの量の両方を測定します。バックアップパフォーマンスを測定する場合、次を考慮します。

- バックアップ手順によって課せられる制限。たとえば、バックアップを 8 時間おきに取得する場合、バックアップが終了するまで 8 時間かからないようにする必要があります。
- ネットワークおよびストレージインフラストラクチャーによって課せられる制限。たとえば、特定のストレージデバイスに多くのバックアップを収納する必要がある場合、バックアッププロセスが遅くなっても、圧縮バックアップを使用する方がよいと考えられます。
- バックアップ時間とリストア時間のトレードオフ。一連のオプションによって、バックアップがやや遅くなるが、リストアを大幅に速くすることができる場合、それらのオプションを選択してもよいと考えられます。リストアプロセスのパフォーマンス情報については、[セクション7.2「リストアパフォーマンスの最適化」](#)を参照してください。

フルまたは増分バックアップ

フルバックアップの取得後、変更されたデータのみがバックアップされる増分バックアップを実行することによって、後続のバックアップを迅速に実行できます。増分バックアップの場合、`mysqlbackup` に、`--incremental` または `--incremental-with-redo-log-only` オプションを指定します。これらのオプションについては、[セクション5.1.8「増分バックアップオプション」](#)を参照してください。増分バックアップのバックアップおよび適用段階の使用方法的指示については、[セクション3.3.2「増分バックアップの作成」](#) および [例4.3「完全バックアップへの増分バックアップの適用」](#)を参照してください。

圧縮バックアップ

バックアップデータを別のサーバーに転送する前に圧縮することは、バックアップが行われるデータベースサーバーへの追加の CPU オーバーヘッドを伴いますが、バックアップデータの最終の宛先となるサーバー上のネットワークラフィックが減少し、ディスク I/O も減少します。圧縮を使用するかどうかを決定する場合、データベースサーバーの負荷、ネットワークの帯域幅、およびデータベースと宛先サーバーの相対容量を考慮します。圧縮バックアップの作成については、[セクション3.3.3「圧縮バックアップの作成」](#) および [セクション5.1.7「圧縮オプション」](#)を参照してください。

圧縮には、バックアップパフォーマンスとリストアパフォーマンスのトレードオフがあります。緊急時に、バックアップデータをリストアする前に圧縮解除するために必要な時間が許容できないことがあります。また、データベースサーバーに圧縮されたバックアップと圧縮解除後のデータの両方を保持するための十分な空き領域がない場合に、ストレージの問題が発生することもあります。そのため、データが重要であるほど、圧縮を使用しないことを選択して、リストアプロセスが可能な限り高速で信頼性が高くなるように、低速で大規模なバックアップを許容することも考えられます。

単一ファイルバックアップ

単一ファイルバックアップ自体は、出力ファイルのディレクトリツリーを生成する従来のバックアップの種類より、必ずしも高速ではありません。そのパフォーマンス上の利点は、それを実行しない場合に、バックアップデータを単一の出カファイルに組み合わせて、それを別のサーバーに転送するなど、連続して実行する必要がありそうなきま

まなステップを組み合わせることにあります。単一ファイルバックアップに関するオプションについては[セクション 5.1.1.5「単一ファイルバックアップの操作」](#)および、使用方法の指示については[セクション 3.3.5「単一ファイルバックアップの作成」](#)を参照してください。

InnoDB 構成オプション設定

MySQL 5.5 以前では、MySQL サーバーが正常にシャットダウンせずに強制終了した場合に、起動時間が長くなるのを避けるため、Redo ログをかなり小さく維持することが一般的な方法でした。MySQL 5.5 以降では、「[InnoDB 構成変数の最適化](#)」に説明するように、[クラッシュリカバリのパフォーマンス](#)が大幅に向上しています。それらのリリースでは、Redo ログファイルを大きくすることが、バックアップ戦略やデータベースワークロードに役立つ場合に、そうすることができます。

後述するように、設定 `innodb_file_per_table=1` で実行することが望ましいと考えられる理由は多数あります。

並列バックアップ

`mysqlbackup` コマンドは最新のマルチコア CPU およびオペレーティングシステムスレッドを利用して、バックアップ操作を並列で実行できます。バックアッププロセスのさまざまな側面に使用されるスレッド数を制御するオプションについては、[セクション 5.1.11「パフォーマンス/スケーラビリティ/容量オプション」](#)を参照してください。バックアップ時に未使用のシステム容量があることがわかっている場合、これらのオプションの値を増加し、そうすることによって、バックアップのパフォーマンスが向上するかどうかをテストすることを考慮してください。

- RAID ストレージ構成を使用して、バックアップパフォーマンスをチューニングし、テストする場合、オプション設定 `--read-threads=3 --process-threads=6 --write-threads=3` の組み合わせを考慮してください。組み合わせ `--read-threads=1 --process-threads=6 --write-threads=1` と比較します。
- 非 RAID ストレージ構成を使用して、バックアップパフォーマンスをチューニングし、テストする場合、オプション設定 `--read-threads=1 --process-threads=6 --write-threads=1` の組み合わせを考慮してください。
- 3つの「スレッド」オプションのいずれかの値を増やす場合は、`--limit-memory` オプションの値も増やして、追加のスレッドに、作業の実行に十分なメモリーを与えます。
- CPU があまりビジーでない (80% 未満の CPU 利用率) 場合は、`--process-threads` オプションの値を増やします。
- バックアップ元のストレージデバイス (ソースドライブ) でもっと多くの I/O リクエストを処理できる場合は、`--read-threads` オプションの値を増やします。
- バックアップ先のストレージデバイス (宛先ドライブ) でもっと多くの I/O リクエストを処理できる場合は、`--write-threads` オプションの値を増やします。

オペレーティングシステムに応じて、`top`、`iostat`、`sar`、`dtrace` などのコマンド、またはグラフィカルパフォーマンスモニターを使用して、リソース利用率を測定できます。システム `iowait` 値が約 20% に達したら、読み取りまたは書き込みスレッド `iowait` の数を増やさないでください。

MyISAM の考慮事項

重要

- `mysqlbackup` コマンドはデータベースの使用を妨げることなく、InnoDB テーブルをバックアップしますが、InnoDB 以外のファイル (MyISAM テーブルや `.frm` ファイルなど) をコピーする最終段階では、ステートメント `FLUSH TABLES WITH READ LOCK` を使用して、一時的にデータベースを読み取り専用状態にします。バックアップパフォーマンスを最高にし、データベース処理への影響を最小にするために:

- バックアップの実行時に長い `SELECT` クエリーやその他の SQL ステートメントを実行しないでください。
- MyISAM テーブルを比較的小さく維持し、おもに読み取り専用または読み取りが大半の作業用にします。

これにより、`mysqlbackup` 実行の最後のロックフェーズが短くなる (おそらく数秒) ため、`mysqld` の通常の処理をあまり妨げません。データベースアプリケーションで先述の

条件が満たされない場合は、`--only-innodb` または `--only-innodb-with-frm` オプションを使用して、InnoDB テーブルのみをバックアップするか、または `--no-locking` オプションを使用して、InnoDB 以外のファイルをバックアップします。`--no-locking` 設定でコピーされる MyISAM、`.frm`、およびその他のファイルがバックアップの最終フェーズで更新されない場合、それらの整合性を保証できないことに注意してください。

- 大きなデータベースの場合、バックアップの実行に長時間かかることがあります。`mysqlbackup` コマンドが終了コード 0 を返していることを確認するか、または `mysqlbackup` がテキスト「mysqlbackup completed OK!」を出力していることを観察して、常に `mysqlbackup` が正常に完了していることをチェックしてください。
- `mysqlbackup` コマンドは、以前の MySQL 6.0 ソースツリーの「MySQL Backup」オープンソースプロジェクトとは異なります。MySQL Enterprise Backup プロジェクトは MySQL Backup イニシアチブに代わりました。
- テーブルに関する DDL 操作が実行中でない期間に、バックアップをスケジュールしてください。DDL 操作と同時のバックアップに対する制約については、[セクション A.1「MySQL Enterprise Backup の制限」](#)を参照してください。

ネットワークパフォーマンス

データ処理操作で、データベースとの通信に Unix ソケットの方が TCP/IP より高速であるという伝統的なアドバイスをご存知かもしれません。`mysqlbackup` コマンドはオプション `--protocol=tcp`、`--protocol=socket`、および `--protocol=pipe` をサポートしていますが、これらのオプションはバックアップやリストアのパフォーマンスに重大な影響を与えません。これらのプロセスには、クライアント/サーバーネットワークトラフィックよりも、ファイルコピー操作がかかります。`--protocol` オプションによって制御されるデータベース通信は少量です。たとえば、`mysqlbackup` はデータベース接続経由でデータベースパラメータに関する情報を取得しますが、テーブルまたはインデックスデータは取得しません。

データサイズ

特定のテーブルやデータベースに重要度の低い情報が格納されているか、めったに更新されない場合、それらをもっと頻繁なバックアップから除外し、もっと少ない頻度のスケジュールでバックアップするようにできます。関連オプションについては、[セクション5.1.9「部分バックアップとリストアオプション」](#)、および特定のテーブル、データベース、またはストレージエンジンからデータを除外することに関する手順については、[セクション3.3.4「部分バックアップの作成」](#)を参照してください。部分バックアップは、少量のデータをコピー、圧縮、転送するため、高速になります。

InnoDB データファイルの全体のサイズを最小にするには、MySQL 構成オプション `innodb_file_per_table` を有効にすることを考慮してください。このオプションは、いくつかの点で、InnoDB テーブルのデータサイズを最小にすることができます。

- これは、InnoDB システムのテーブルスペースのサイズの拡大を防ぎ、後で MySQL によってのみ使用可能なディスク領域を割り当てます。たとえば、大量のデータが一時的にのみ必要になったり、誤ってまたは試験中にロードされたりすることがあります。`innodb_file_per_table` オプションを使用しないと、システムテーブルスペースがこのすべてのデータを保持するために拡張し、その後縮小することはありません。
- これは、テーブルが削除されるか、切り捨てられると、InnoDB テーブルとそのインデックスによって使用されているディスク領域をすぐに解放します。各テーブルとその関連付けられたインデックスは、これらの DDL 操作によって削除されるか、空にされる `.ibd` ファイルによって表されます。
- これにより、大量のデータが削除されるか、インデックスが削除された場合に、`.ibd` ファイル内の未使用の領域が `OPTIMIZE TABLE` ステートメントによって回収されます。
- これにより、[セクション3.3.4「部分バックアップの作成」](#)に説明するように、一部の InnoDB テーブルをバックアップし、その他をバックアップしない部分バックアップが可能になります。

クエリーで使用されていないインデックスの作成を避けます。インデックスはバックアップデータの領域を占めるため、不要なインデックスによってバックアッププロセスが遅くなります。(mysqlbackup によって使用されるコピーおよびスキャンメカニズムでは、それらの作業の実行にインデックスに依存しません。)たとえば、クエリーによって使用されるインデックスは 1 つだけであるため、テーブルのカラムごとにインデックスを作成しても一般に役立ちませ

ん。プライマリキーカラムは各 InnoDB セカンダリインデックスに含まれるため、多数または著しく長いカラムや、同じカラムのさまざまな配列の複数のセカンダリインデックスから構成されるプライマリキーを定義する領域を無駄にします。

Apply-Log フェーズ

別個のマシンにバックアップデータを格納し、そのマシンがデータベースサーバーをホストするマシンほどビジネスでない場合、一部の後処理の作業 (apply-log フェーズ) をその別個のマシンにオフロードできます。[セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」](#)

初期バックアップ後すぐに apply-log フェーズを実行する (リストアを高速にする) か、またはリストア直前まで延期する (バックアップを高速にする) には、常にパフォーマンスのトレードオフがあります。緊急時に、リストアパフォーマンスはもっとも重要な考慮事項です。そのため、データが重要であるほど、apply-log フェーズをバックアップ直後に実行することが重要になります。[backup-and-apply-log](#) オプションを指定して、同じサーバーでバックアップフェーズと apply-log フェーズを組み合わせるか、または高速の初期バックアップを実行し、バックアップデータを別のサーバーに転送してから、[セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」](#) のいずれかのオプションを使用して、apply-log フェーズを実行します。

7.2 リストアパフォーマンスの最適化

このセクションでは、MySQL Enterprise Backup 製品によるリストア操作のパフォーマンスの考慮事項について説明します。このサブジェクトは、次の理由で、バックアップパフォーマンスの議論と別に、独自のセクションに値するほど重要です。

- リストア操作は、バックアップ方法の違いによってきわめて大きく異なる傾向があるバックアップとリストアサイクルのフェーズです。たとえば、[mysqldump](#) を使用したバックアップパフォーマンスが許容できても、[mysqldump](#) では一般にリストア操作に MySQL Enterprise Backup よりはるかに時間がかかります。
- リストア操作は多くの場合、アプリケーションや Web サイトの停止時間を最小にすることがきわめて重要である緊急時に実行されます。
- リストア操作は、常にデータベースサーバーがシャットダウンした状態で実行されます。
- リストア操作は、ファイルを転送する場合の I/O とネットワーク速度、およびデータを圧縮解除する場合の CPU 速度、プロセッサコアなどの低レベルの考慮事項にも依存します。

リストアジョブに指定できるオプションの組み合わせについては、[セクション 5.1.1.3 「既存のバックアップのリストア」](#) を参照してください。

さまざまなクラスのバックアップデータのリストア

部分バックアップのリストアは、物理的にコピーするデータが少ないため、フルバックアップのリストアより時間がかかりません。部分バックアップの作成については、[セクション 5.1.9 「部分バックアップとリストアオプション」](#) を参照してください。

圧縮バックアップのリストアは、データの圧縮解除に必要な時間が、一般にネットワーク経由で転送されるデータが少ないことによって節約される時間より大きくなるため、非圧縮バックアップのリストアより時間がかかります。バックアップをリストアする前に、ストレージを再整理して、バックアップを圧縮解除するための十分な領域を解放する必要がある場合、必要な合計時間の見積もりに、その管理作業を含めます。緊急時に、バックアップデータをリストアする前に圧縮解除するために必要な時間が許容できないことがあります。データベースサーバーに圧縮されたバックアップと圧縮解除後のデータの両方を保持するための十分な空き領域がない場合、そのため、データが重要であるほど、圧縮を使用しないことを選択して、リストアプロセスが可能な限り高速で信頼性が高くなるように、低速で大規模なバックアップを許容することも考えられます。圧縮バックアップの作成については、[セクション 5.1.7 「圧縮オプション」](#) を参照してください。

単一ファイルバックアップをリストアするためのアンパッキングプロセスは、一般に実効速度や余分なストレージの点で、コストがかかりません。各ファイルは、その最終の宛先に直接アンパックされ、個別にコピーされた場合と同じです。そのため、単一ファイルバックアップを使用することで、バックアップを大幅に高速化したり、そのストレージ要件を削減したりできる場合、それは一般にリストア時間とのトレードオフが発生しません。単一ファイルバックアップの作成については、[セクション 5.1.1.5 「単一ファイルバックアップの操作」](#) を参照してください。

Apply-Log フェーズ

別個のマシンにバックアップデータを格納し、そのマシンがデータベースサーバーをホストするマシンほどビジーでない場合、一部の後処理の作業 (apply-log フェーズ) をその別個のマシンにオフロードできます。セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」

初期バックアップ後すぐに apply-log フェーズを実行する (リストアを高速にする) か、またはリストア直前まで延期する (バックアップを高速にする) には、常にパフォーマンスのトレードオフがあります。緊急時に、リストアパフォーマンスはもっとも重要な考慮事項です。そのため、データが重要であるほど、apply-log フェーズをバックアップ直後に実行することが重要になります。backup-and-apply-log オプションを指定して、同じサーバーでバックアップフェーズと apply-log フェーズを組み合わせるか、または高速の初期バックアップを実行し、バックアップデータを別のサーバーに転送してから、セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」のいずれかのオプションを使用して、apply-log フェーズを実行します。

セクション 5.1.1.2 「既存のバックアップデータの apply-log 操作」

ネットワークパフォーマンス

データ処理操作で、データベースとの通信に Unix ソケットの方が TCP/IP より高速であるという伝統的なアドバイスをご存知かもしれません。mysqlbackup コマンドはオプション `--protocol=tcp`、`--protocol=socket`、および `--protocol=pipe` をサポートしていますが、これらのオプションはバックアップやリストアのパフォーマンスに重大な影響を与えません。これらのプロセスには、クライアント/サーバーネットワークトラフィックよりも、ファイルコピー操作がかかわります。`--protocol` オプションによって制御されるデータベース通信は少量です。たとえば、mysqlbackup はデータベース接続経由でデータベースパラメータに関する情報を取得しますが、テーブルまたはインデックスデータは取得しません。

並列リストア

mysqlbackup コマンドは最新のマルチコア CPU およびオペレーティングシステムスレッドを利用して、バックアップ操作を並列で実行できます。リストアプロセスのさまざまな側面に使用されるスレッド数を制御するオプションについては、セクション 5.1.11 「パフォーマンス/スケーラビリティ/容量オプション」を参照してください。リストア時に未使用のシステム容量があることがわかっている場合、これらのオプションの値を増加し、そうすることによって、リストアのパフォーマンスが向上するかどうかをテストすることを考慮してください。

- RAID ストレージ構成を使用して、バックアップパフォーマンスをチューニングし、テストする場合、オプション設定 `--read-threads=3 --process-threads=6 --write-threads=3` の組み合わせを考慮してください。組み合わせ `--read-threads=1 --process-threads=6 --write-threads=1` と比較します。
- 非 RAID ストレージ構成を使用して、バックアップパフォーマンスをチューニングし、テストする場合、オプション設定 `--read-threads=1 --process-threads=6 --write-threads=1` の組み合わせを考慮してください。
- 3 つの「スレッド」オプションのいずれかの値を増やす場合は、`--limit-memory` オプションの値も増やして、追加のスレッドに、作業の実行に十分なメモリーを与えます。
- CPU があまりビジーでない (80% 未満の CPU 利用率) 場合は、`--process-threads` オプションの値を増やします。
- リストア元のストレージデバイス (ソースドライブ) でもっと多くの I/O リクエストを処理できる場合は、`--read-threads` オプションの値を増やします。
- リストア先のストレージデバイス (宛先ドライブ) でもっと多くの I/O リクエストを処理できる場合は、`--write-threads` オプションの値を増やします。

オペレーティングシステムに応じて、`top`、`iostat`、`sar`、`dtrace` などのコマンド、またはグラフィカルパフォーマンスモニターを使用して、リソース利用率を測定できます。システム `iowait` 値が約 20% に達したら、読み取りまたは書き込みスレッド `iowait` の数を増やさないでください。

第 8 章 バックアップの暗号化

バックアップされたデータのセキュリティを高めるため、バージョン 3.10,以降、MySQL Enterprise Backup では単一ファイルバックアップの暗号化を提供しています。暗号化は、部分、圧縮、または増分単一ファイルバックアップを作成する場合、およびバックアップデータを別のデバイスやサーバーにストリーミングする場合にも適用できます。

暗号化は、ユーザーによって指定された 64 桁の 16 進数の鍵文字列を使用して、CBC モードの Advanced Encryption Standard (AES) ブロック暗号化によって実行されます。復号化は同じ鍵を使用して実行されます。鍵は、単に 64 個のランダムな 16 進バイトを組み合わせて手動で作成するか、または `shasum` (またはプラットフォームで機能するハッシュ計算用の類似のプログラム) にキープレーズを指定して生成することができます。

```
$ echo -n "my secret passphrase" | shasum -a 256
a7e845b0854294da9aa743b807cb67b19647c1195ea8120369f3d12c70468f29 -
```

末尾の「-」は鍵に含まれず、無視すべきであることに注意してください。鍵を `--key` オプションで `mysqlbackup` に提供するか、鍵ファイルに鍵を張り付けて、ファイルのパス名を `--key-file` オプションで `mysqlbackup` に提供します。

鍵をランダムに生成するには、OpenSSL などのツールを使用できます。

```
$ openssl rand 32 -hex
8f3ca9b850ec6366f4a54feba99f2dc42fa79577158911fe8cd641ffff1e63d6
```

OpenSSL によって生成された鍵を鍵ファイルに入れるには、次を実行します。

```
$ openssl rand 32 -hex >keyfile
$ cat keyfile
6a1d325e6ef0577f3400b7cd624ae574f5186d0da2eeb946895de418297ed75b
```

暗号化機能では MySQL Enterprise Backup の独自の暗号化形式を使用しており、復号化は MySQL Enterprise Backup を使用することによってのみ可能です。Unix のようなオペレーティングシステムでは、暗号化されたバックアップファイルと暗号化されていないバックアップファイルを識別するために異なるマジック値が使用されます。たとえば、これらの行をオペレーティングシステムの `/etc/magic` ファイルに追加できます。

```
0 string MBackuP\n MySQL Enterprise Backup backup image
0 string MebEncR\n MySQL Enterprise Backup encrypted backup
```

その後、`file` コマンドを使用して、ファイルの種類を識別できます。

```
$ file /backups/image1 /backups/image2
/backups/image1: MySQL Enterprise Backup backup image
/backups/image2: MySQL Enterprise Backup encrypted backup
```

暗号化および復号化に使用するコマンドオプションは `--encrypt`、`--decrypt`、`--key`、および `--key-file` です。これらのオプションは、バックアップイメージへのさまざまな操作で使用できます。詳細は、[セクション 5.1.14 「暗号化オプション」](#)を参照してください。

次は、暗号化バックアップを作成するためのサンプルコマンドです。

```
mysqlbackup --backup-image=/backups/image.enc --encrypt
--key=23D987F3A047B475C900127148F9E0394857983645192874A2B3049570C12A34
--backup-dir=/var/tmp/backup backup-to-image
```

同じタスクに鍵ファイルを使用するには:

```
mysqlbackup --backup-image=/backups/image.enc --encrypt
--key-file=/meb/key --backup-dir=/var/tmp/backup backup-to-image
```

バックアップを抽出時に復号化するには:

```
mysqlbackup --backup-image=/backups/image.enc --decrypt
--key-file=/meb/key --backup-dir=/backups/extract-dir extract
```

暗号化されたバックアップイメージを検証するには:

```
mysqlbackup --backup-image=/logs/encimage.bi --decrypt --key-file=/meb/enckey validate
```

第 9 章 MySQL Enterprise Backup をメディア管理ソフトウェア (MMS) 製品と使用する

目次

9.1 Oracle Secure Backup によるテープへのバックアップ	109
---	-----

このセクションでは、メディア管理ソフトウェア (MMS) 製品と組み合わせて MySQL Enterprise Backup を使用方法について説明します。それらの製品は、多くの場合にテープドライブなどの大容量バックアップデバイスとともに、一般に大量のバックアップデータを管理するために使用します。

9.1 Oracle Secure Backup によるテープへのバックアップ

テープドライブは、バックアップデータ用の価格が手頃な大容量のストレージデバイスです。MySQL Enterprise Backup 製品は、Oracle Secure Backup (OSB) などのメディア管理ソフトウェア (MMS) とやり取りして、MySQL バックアップおよびリストアジョブを進めます。メディア管理ソフトウェアは、バージョン 2 以上の System Backup to Tape (SBT) インタフェースをサポートしている必要があります。

MySQL Enterprise Backup 側で、ファイル名の前にプリフィクス `sbt:` を付け、`--backup-image` パラメータを使用して、単一ファイルバックアップとしてバックアップジョブを実行し、オプションでほかの `--sbt-*` パラメータを `mysqlbackup` コマンドに渡して、SBT 処理のさまざまな側面を制御します。`--sbt-*` オプションを [セクション 5.1.10 「単一ファイルバックアップオプション」](#) に示しています。

OSB 側で、`mysqlbackup` を呼び出す構成可能なコマンドを指定して、MySQL Enterprise Backup ジョブをスケジュールできます。暗号化などの OSB の機能を制御するには、それらの機能を特定のバックアップに適用する「ストレージセクタ」を定義し、MySQL Enterprise Backup パラメータ `--sbt-database-name=storage_selector` を使用して、ストレージセクタの名前を OSB に渡します。

MySQL データをテープにバックアップするには:

- `mysqlbackup` コマンドの `--backup-image=sbt:name` パラメータを指定して、バックアップデータを一意に識別します。`sbt:` プリフィクスは、バックアップデータをローカルファイルではなく、MMS に送信し、引数値の残りは、MMS 内の一意のバックアップ名として使用されます。
- `mysqlbackup` コマンドの `--sbt-database-name` パラメータを指定し、OSB オペレータがこの MySQL ソースからバックアップ用のストレージセクタを構成できるようにします。(このパラメータは OSB オペレータによって定義された「ストレージセクタ」を表し、MySQL データベース名ではありません。)デフォルトで、`mysqlbackup` はこの MMS パラメータに `MySQL` の値を指定します。このオプションへの引数は 8 バイトに制限されます。
- 複数のメディア管理プログラムがインストールされている場合、使用する特定の SBT ライブラリを選択するには、`mysqlbackup` コマンドの `--sbt-lib-path` パラメータを指定します。`--sbt-lib-path` パラメータを指定しない場合、`mysqlbackup` は通常の実行システムパスと環境変数を使用して、Linux および Unix システムでは `libobk.so` および Windows システムでは `ORASBT.DLL` という名前の SBT ライブラリを見つけます。`--sbt-lib-path` を指定する場合、パスの指定に加えて、ライブラリに別のファイル名を使用することができます。
- 通常 `--sbt-environment` オプションを使用して、環境変数によって制御されるその他の製品固有の設定を指定します。

テープから MySQL データをリストアするには:

- リストア操作の一部として `mysqlbackup` コマンドの `--backup-image=sbt:name` パラメータを指定します。元のバックアップ時と同じ `name` 値を使用します。この単一のパラメータは、該当するテープデバイスから該当するデータを取得します。
- オプションで、バックアップ操作の場合と同じ値を使用して、`--sbt-lib-path` オプションを使用します。
- 通常 `--sbt-environment` オプションを使用して、環境変数によって制御されるその他の製品固有の設定を指定します。

Oracle Secure Backup の製品固有の情報については、[Oracle Secure Backup のドキュメント](#)を参照してください。

例 9.1 Oracle Secure Backup と MySQL Enterprise Backup を使用したサンプル `mysqlbackup` コマンド

```
# Uses libobk.so or ORASBT.DLL in standard places):
mysqlbackup --port=3306 --protocol=tcp --user=root --password \
--backup-image=sbt:backup-shoeprod-2011-05-30 \
--backup-dir=/backup backup-to-image

# Associates this backup with storage selector 'shoeprod':
mysqlbackup --port=3306 --protocol=tcp --user=root --password \
--backup-image=sbt:backup-shoeprod-2011-05-30 \
--sbt-database-name=shoeprod \
--backup-dir=/backup backup-to-image

# Uses an alternative SBT library, /opt/Other-MMS.so:
mysqlbackup --port=3306 --protocol=tcp --user=root --password \
--backup-image=sbt:backup-shoeprod-2011-05-30 \
--sbt-lib-path=/opt/Other-MMS.so \
--backup-dir=/backup backup-to-image
```

第 10 章 MySQL Enterprise Backup のトラブルシューティング

目次

10.1 MySQL Enterprise Monitor によるバックアップのモニタリング	111
10.2 MySQL Enterprise Backup のエラーコード	111
10.3 破損の問題の回避	111
10.4 MySQL Enterprise Backup ログの使用	112
10.5 MySQL Enterprise Backup マニフェストの使用	114

MySQL Enterprise Backup 製品によるバックアップとリストアに関する問題をトラブルシューティングするには、次の点を考慮してください。

- 何らかの問題のトラブルシューティングの前に、[付録A「MySQL Enterprise Backup の制限」](#)で製品の既知の制限と制約を理解しておいてください。
- `mysqlbackup` が、オペレーティングシステムの呼び出し時に問題を検出した場合、対応する OS エラーコードを返します。これらのエラーコードの意味と解決方法については、オペレーティングシステムのドキュメントを参照する必要があります。
- `mysqlbackup` コマンドからの出力は `stdout` ではなく `stderr` に送信されます。デフォルトで、同じ出力がエラー診断に使用される `backup_dir` 内のログファイルにも保存されます。このロギング機能の構成方法の詳細については、[セクション5.1.12「メッセージロギングオプション」](#)を参照してください。
- 増分バックアップでは、一連の期間を指定する場合に注意が必要です。各バックアップの最後に、最終 LSN 値を記録し、その値を次の増分バックアップで指定する必要があります。さらに、一連の増分バックアップからのすべての変更が含まれるように、まずリストアするフルバックアップが正しく準備されていることを確認する必要があります。
- `mysqlbackup` コマンドの進行に従って、進行状況情報が `mysql.backup_progress` テーブルに書き込まれます。コマンドは、バックアップ操作を終了すると、`mysql.backup_history` テーブルにステータス情報を記録します。これらのテーブルをクエリーして、進行中のジョブをモニターし、さまざまな段階で必要とされた時間を確認して、エラーが発生していないかどうかをチェックできます。

10.1 MySQL Enterprise Monitor によるバックアップのモニタリング

MySQL Enterprise Backup 製品と MySQL Enterprise Monitor 製品の組み合わせによって、独自のクエリーやスクリプトを書かなくても、バックアップジョブの進行状況と履歴をモニターできます。

- MySQL Enterprise Monitor のグラフ `Backup Run Time` と `Backup Locked Time` はバックアップジョブのフェーズでかかる時間を図に示します。
- MySQL Enterprise Monitor のルール `MySQL Enterprise Backup Failed`、`MySQL Enterprise Backup Succeeded`、`MySQL Enterprise Backup Lock Time Excessive`、`Incremental MySQL Enterprise Backups Not Enabled`、および `Last Full MySQL Enterprise Backup Too Old` はバックアップジョブに関連する問題を警告します。

モニタリング機能には、MySQL Enterprise Backup 3.5.3 以上および MySQL Enterprise Monitor 2.3.4 以上が必要です。これらの MySQL Enterprise Monitor の機能については、[MySQL Enterprise Monitor のユーザズガイド](#)を参照してください。

10.2 MySQL Enterprise Backup のエラーコード

バックアップまたはリストアの実行が成功した場合、MySQL Enterprise Backup (`mysqlbackup`) プロセスのリターンコードは 0 になります。実行が何らかの理由で失敗した場合、リターンコードは 1 になります。

10.3 破損の問題の回避

オペレーティングシステムやハードウェアによって、データベースエラーにならなくても、`mysqlbackup` の実行を妨げる場所で、データファイルページを破損させることがあります。

```
mysqlbackup: Re-reading page at offset 0 3185082368 in /sqldata/mts/ibdata15
bbackup: Re-reading page at offset 0 3185082368 in /sqldata/mts/ibdata15
bbackup: Error: page at offset 0 3185082368 in /sqldata/mts/ibdata15 seems corrupt!
```

破損の問題にはさまざまな原因がある可能性があります。ここに、その処理に関するいくつかの提案を示します。

- この問題は、MySQL サーバーが著しくビジーな場合に発生する可能性があります。ほかの解決方法を試してみる前に、次の `mysqlbackup` オプションのデフォルト以外の設定を使用して、バックアップを再度実行した方がよい場合があります。
- `--page-reread-time=MS`。たとえば、チェックサムの失敗時に、再読み取りを速くするために、値を「0.05」に設定してみます。
- `--page-reread-count=retry_limit`。たとえば、チェックサムの失敗時に、MySQL Enterprise Backup が中断し、エラーを生成するまでの再読み取りを増やすために、値を「1000」に設定してみます。
- メモリー内で混乱したデータによって、ディスク上のデータが実際に破損していても、問題が発生することがあります。データベースサーバーとストレージデバイスをリブートし、問題が続くかどうかを確認します。
- データベースサーバーとストレージデバイスを再起動した後に、問題が続く場合は、実際にディスクに破損がある可能性があります。以前のバックアップからデータをリストアし、最近の変更を「巻き戻し」して、データベースをその最新状態に戻すことを考慮できます。
- 問題の原因を調査する前に、MySQL Enterprise Backup でとにかくバックアップを終了させたい場合は、サーバーで `innochecksum` ユーティリティを実行して、ディスク上のチェックサム値を書き換えることができます。

```
innochecksum --no-checksum --write=crc32
```

オプション `--no-checksum` はツールの検証機能を無効にし、オプション `--write=crc32` は `innochecksum` にディスク上のチェックサム値を書き換えさせます。

重要: 破損の問題を小さな問題として扱わないでください。破損が発生しているシステムでどこが悪いのか調査しますが、そうしたトラブルシューティングはこのマニュアルの範囲を超えています。

10.4 MySQL Enterprise Backup ログの使用

`mysql.backup_progress` テーブルにより、バックアップジョブの実行時にそれらをモニターできます。`mysql.backup_history` テーブルにより、完了したジョブの結果を確認できます。これらのテーブルは CSV ストレージエンジンによって作成されているため、それらを SQL からクエリーしたり、アプリケーションまたはスクリプトからテキストファイルを解析したりすることができます。

バックアップ操作でこれらのテーブルの更新をスキップするには、`--no-history-logging` オプションを使用します。

backup_progress テーブル

`backup_progress` テーブルの各行は、実行中のバックアップジョブからの状態の変更やメッセージを記録します。`backup_progress` テーブルには次のカラムがあります。

- `backup_id`
- `tool_name`
- `error_code`
- `error_message`
- `current_time`
- `current_state`

CSV ストレージエンジンは直接 `NULL` 値を表現できないため、ログでは代わりに `-1` の値が使用されます (たとえば、バイナリロギングが有効にされていない場合に、`binlog_pos` カラムなどで)。

`backup_id` 値は、単一のバックアップ操作の情報をグループ化し、ジョブの終了後に `backup_history` テーブル内の対応する行に関連付けるために使用します。

`error_code` 値と `error_message` 値は、ジョブの進行状況を追跡し、バックアップ操作を停止する必要がある重大なエラーが発生したかどうかを検出するために使用します。

`current_time` 値と `current_state` 値は、バックアップ操作の各部分でかかる時間を測定し、将来のバックアップの間隔の計画に役立てるために使用します。

backup_history テーブル

`backup_history` テーブルの各行は、`mysqlbackup` コマンドによって生成された、1 つの完了したバックアップジョブの詳細を記録します。`backup_history` テーブルには次のカラムがあります。

- `backup_id`
- `tool_name`
- `start_time`
- `end_time`
- `binlog_pos`
- `binlog_file`
- `compression_level`
- `engines`
- `innodb_data_file_path`
- `innodb_file_format`
- `start_lsn`
- `end_lsn`
- `backup_type`
- `backup_format`
- `mysql_data_dir`
- `innodb_data_home_dir`
- `innodb_log_group_home_dir`
- `innodb_log_files_in_group`
- `innodb_log_file_size`
- `backup_destination`
- `lock_time`
- `exit_state`
- `last_error`
- `last_error_code`

`end_lsn` 値は、増分バックアップに関連する操作を自動化するために使用します。フルバックアップまたは増分バックアップを取得した場合、そのバックアップからの終了 LSN を次の増分バックアップの開始 LSN として指定します。

`mysql_data_dir`、`innodb_data_home_dir`、および `backup_destination` などのバックアップ関連の構成設定に対応する値は、バックアップで正しいソースディレクトリと宛先ディレクトリを使用していることを確認するために使用します。

値 `exit_state`、`last_error`、および `last_error_code` は、各バックアップの成功または失敗を評価するために使用します。

`last_error` が 'NO_ERROR' の場合、バックアップ操作は成功しています。何らかのエラーがある場合、`backup_progress` テーブルから、そのバックアップ操作のエラーの完全なリストを取得できます。

10.5 MySQL Enterprise Backup マニフェストの使用

各バックアップディレクトリには、`meta` サブディレクトリに、バックアップが生成された状態およびそれに格納されているファイルの詳細を示すいくつかのファイルが含まれます。この情報を含むファイルは、集合的に **マニフェスト** と呼ばれます。

`mysqlbackup` は、データベース管理ツールによって使用されるために、これらのファイルを生成し、それらの作成後、マニフェストファイルを参照したり、変更したりすることはありません。管理ツールでは、診断およびトラブルシューティング手順でマニフェストを使用できます。たとえば、元の MySQL インスタンスが完全に失われ、リカバリプロセスが稼働中の MySQL サーバーにファイルをコピーするよりもかなりの量になる場合などです。

マニフェスト内のファイルには次が含まれます。

- `backup_create.xml`: バックアップ操作に関する情報。
- `backup_content.xml`: バックアップ内のファイルに関する情報。この情報は、バックアップ操作が成功した場合にのみ完成され、整合性があります。このファイルの内容は将来拡張される可能性があります。管理ツールはこの情報を使用して、フルバックアップまたは `--databases` オプションによって実行された部分バックアップに含まれるテーブルを確認します。(この情報は、`--include`、`--incremental`、`--incremental-with-redo-log-only`、`--only-innodb`、または `--only-innodb-with-frm` オプションで取得された部分バックアップには存在しません。)管理ツールでは、単一ファイルバックアップのマニフェストに記録されたチェックサムを、単一ファイルバックアップのアンパック後のファイルのチェックサムと比較することができます。ファイルには、バックアップ対象サーバーで定義されているすべてのプラグインの詳細も含まれます。ユーザーはこれにより、同じプラグインがリストアのターゲットサーバーでも同じように定義されていることを確認する必要があります。
- `image_files.xml`: 単一ファイルバックアップ内のファイルに関する情報。(`backup-to-image` オプションと `backup-dir-to-image` オプションによって取得されたバックアップの場合にのみ生成されます。)管理ツールでは、このファイルに記録されたパスを使用し、`image-to-backup-dir` オプションまたは `extract` オプションを使用して、単一ファイルバックアップのアンパックを計画または自動化したり、または `--src-entry` および `--dst-entry` オプションによって、抽出されたファイルのパスを再配置したりすることができます。

第 11 章 MySQL Enterprise Backup のよくある質問

このセクションでは、MySQL Enterprise Backup に関する一般的な質問を、回答と詳細情報のポインターとともに示します。

11.1 MySQL Enterprise Backup は MySQL Server バージョン <code>x.y.z</code> で動作しますか。	115
11.2 すべてのバックアップにある大きな <code>ibdata</code> ファイルは何ですか。	115
11.3 MySQL Enterprise Backup で InnoDB 以外のデータをバックアップできますか。	115
11.4 「apply」ステップが中断されると何が起こりますか。	115
11.5 オプション <code>--defaults-file</code> が認識されないのはなぜですか。	115
11.6 MySQL Enterprise Backup を使用して、ある OS プラットフォームでデータベースをバックアップし、それを別のプラットフォームでリストアできますか。	115
11.7 Windows プラットフォームで、MySQL サーバーのインストーラを使用して、MySQL Enterprise Backup 3.8.2 を 3.9.0 にアップグレードできますか。	116

11.1. MySQL Enterprise Backup は MySQL Server バージョン `x.y.z` で動作しますか。

MySQL Enterprise Backup と MySQL Server のさまざまなリリース間の互換性の詳細については、[セクション B.3「古い MySQL バージョンとの互換性」](#)を参照してください。

11.2. すべてのバックアップにある大きな `ibdata` ファイルは何ですか。

`ibdata1` などの名前が付いた大きなファイルのため、バックアップデータが予想より多くの領域を占めていることに気付くことがあります。このファイルは InnoDB [システムテーブルスペース](#)を表し、これは拡大しても縮小することはなく、すべてのフルバックアップおよび増分バックアップに含まれます。バックアップデータ内のこのファイルによって占められる領域を削減するには:

- [フルバックアップ](#)の実行後、占有する領域が少ない[増分バックアップ](#)を連続して実行します。増分バックアップ内の `ibdata1` ファイルは、フルバックアップから変更されたシステムテーブルスペースの部分のみが含まれ、一般にはるかに小さくなります。
- 最大またはもっともアクティブな InnoDB テーブルを作成する前に、構成オプション `innodb_file_per_table=1` を設定します。それらのテーブルは、システムテーブルスペースから個別の `.ibd` ファイルに分割されます。これらは、削除されたり、切り捨てられたりした場合に、ディスク領域を解放するという点で、柔軟性が高く、バックアップに個別に含めたり、除外したりできます。
- `innodb_file_per_table` 設定をオンにする前に、大容量の InnoDB データを作成したため、システムテーブルスペースが著しく大きくなった場合、すべてのテーブルデータがシステムテーブルスペース外で維持されるように、[mysqldump](#) を使用して、インスタンス全体をダンプしてから、それを再作成する前に、`innodb_file_per_table` をオンにすることができます。

11.3. MySQL Enterprise Backup で InnoDB 以外のデータをバックアップできますか。

MySQL Enterprise Backup では InnoDB 以外のデータ (MYISAM テーブルなど) をバックアップできますが、バックアップする MySQL サーバーで InnoDB をサポートしている必要があります (つまり、サーバーが `--innodb=OFF` または `--skip-innodb` オプションを使用して起動された場合、バックアッププロセスは失敗します)、サーバーに少なくとも 1 つの InnoDB テーブルが含まれている必要があります。

11.4. 「apply」ステップが中断されると何が起こりますか。

`mysqlbackup` コマンドが `apply-log` または `apply-incremental-backup` ステージで中断された場合、バックアップデータは問題ありません。それらのオプションによって実行されるファイル操作は、バックアップデータの整合性を損なわずに、何回でも実行できます。同じ `mysqlbackup` コマンドを再度実行し、それが正常に完了すると、必要なすべての変更がバックアップデータに存在します。

11.5. オプション `--defaults-file` が認識されないのはなぜですか。

`--defaults-file` オプションを指定する場合、コマンド名の後の先頭のオプションにする必要があります。そうしないと、エラーメッセージにより、そのオプション名が認識されていないかのように見えます。

11.6. MySQL Enterprise Backup を使用して、ある OS プラットフォームでデータベースをバックアップし、それを別のプラットフォームでリストアできますか。

詳細は、[セクションB.1「プラットフォーム間の互換性」](#)を参照してください。

- 11.7. Windows プラットフォームで、MySQL サーバーのインストーラを使用して、MySQL Enterprise Backup 3.8.2 を 3.9.0 にアップグレードできますか。

Windows プラットフォームで、MySQL Enterprise Backup 3.8.2 がすでにインストールされており、MySQL Enterprise Backup 3.9.0 もインストールする MySQL サーバーのインストーラを実行した場合、MySQL インストーラは、MySQL Enterprise Backup 3.8.2 を 3.9.0 にアップグレードするオプションを提供しません。それらは別の製品とみなされているためです。インストーラは MySQL Enterprise Backup 3.9.0 をインストールし、3.8.2 を変更しないままにします。

第 III 部 付録

目次

A MySQL Enterprise Backup の制限	121
A.1 MySQL Enterprise Backup の制限	121
B MySQL Enterprise Backup の互換性情報	123
B.1 プラットフォーム間の互換性	123
B.2 古い MySQL Enterprise Backup との互換性	123
B.3 古い MySQL バージョンとの互換性	123
B.4 特定の MySQL バージョンに関する互換性の注意事項	123
C 詳細な例	125
C.1 完全バックアップのディレクトリ構造例	125
C.2 圧縮バックアップのディレクトリ構造例	129
C.3 増分バックアップのディレクトリ構造例	129
D MySQL Enterprise Backup リリースノート	131
E サードパーティーコンポーネントライセンス	133
E.1 RegEX-Spencer ライブラリライセンス	133
E.2 zlib ライセンス	133
E.3 LZ4 ライセンス	134
E.4 Percona Multiple I/O スレッドパッチライセンス	134
E.5 Google SMP Patch ライセンス	135
E.6 Google Controlling Master Thread I/O Rate Patch ライセンス	135
E.7 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) ライセンス	136
E.8 OpenSSL v1.0 ライセンス	136
E.9 cURL (libcurl) ライセンス	138
MySQL Enterprise Backup 用語集	139

付録 A MySQL Enterprise Backup の制限

目次

A.1 MySQL Enterprise Backup の制限	121
---------------------------------------	-----

`mysqlbackup` の修正済みバグの一覧については、付録D「MySQL Enterprise Backup リリースノート」で MySQL Enterprise Backup のバージョン履歴を参照してください。

A.1 MySQL Enterprise Backup の制限

- MySQL 5.6 以降のグループコミット機能により、InnoDB Redo ログのフラッシュ操作の頻度が変わるため、InnoDB テーブルからのバックアップデータに関連付けられた特定の時点に影響が及ぶ可能性があります。詳細は、[セクションB.4「特定の MySQL バージョンに関する互換性の注意事項」](#)を参照してください。
- MySQL 5.5 以前では、[セクション4.4「単一の .ibd ファイルのバックアップとリストア」](#)で述べているように、個々の InnoDB テーブルをリストアするときには、バックアップ後に MySQL Server でテーブルが破棄または切り取られていないことが必要です。InnoDB テーブルを破棄したり切り取りたりすると内部テーブル ID が変わるため、テーブルが再作成されると、その ID はバックアップデータからのテーブル ID と一致しくなくなります。同じシリーズの MySQL Server の、ある一般提供開始 (GA) バージョンから別バージョンにリストアが行われるかぎり、この制限は MySQL 5.6 以降には適用されません。
- Linux、Unix、および OS X システムで、`mysqlbackup` コマンドは、バックアップされるファイルのファイル所有権もアクセス許可も記録しません。リストア時、たとえば `mysql` ではなく `root` が所有しているなど、これらのファイルの所有権が異なる場合があります。また、たとえばファイル所有者だけではなく全員が読み取り可能であるなど、読み取り/書き込みアクセス許可が異なる場合もあります。バックアップ戦略を計画するときに、MySQL データディレクトリ内のファイルを調べて、それらの所有者とアクセス許可設定が一貫していることを確認してください。リストア操作を実行するときには、`su`、`umask`、`chown`、および `chmod` の適切な組み合わせを使用して、元のファイルと同じ所有者および権限を設定します。
- 場合によっては、MyISAM テーブルなどの非トランザクションテーブルのバックアップには、コミットされていない追加データが含まれることがあります。`autocommit` をオフにし、InnoDB テーブルと非トランザクションテーブルの両方を同じトランザクション内で変更した場合、バイナリログ位置が更新される前に、非トランザクションテーブルにデータを書き込みます。トランザクションがコミットされるときにバイナリログ位置が更新されますが、非トランザクションデータは即座に書き込まれます。このようなトランザクションが開かれている間にバックアップが行われた場合、バックアップデータには、非トランザクションテーブルに行われた更新が含まれます。
- Unix `kill -9` コマンドなどで `mysqlbackup` プロセスが中断された場合、`FLUSH TABLES WITH READ LOCK` 操作が実行し続けることがあります。この場合、`mysql` コマンド行から `KILL QUERY` ステートメントを使用して、`FLUSH TABLES WITH READ LOCK` ステートメントを強制終了してください。長時間実行するクエリーまたはトランザクションによって、`FLUSH TABLES` 操作が停止した場合に、この問題が起きる可能性が高くなります。バックアップのタイミングとパフォーマンスに関するガイドラインについては、[第5章「mysqlbackup コマンドリファレンス」](#)を参照してください。
- バックアップ操作の進行中は、`ALTER TABLE`、`TRUNCATE TABLE`、`OPTIMIZE TABLE`、`REPAIR TABLE`、および `RESTORE TABLE` の DDL 操作を実行しないでください。結果として得られたバックアップが破損している可能性があります。

バックアップと並列して安全に実行できる `ALTER TABLE` 操作だけが、変化するカラム名やデフォルトのカラム値など、ディスク上のレコードの物理表現に影響しない操作です。
- MySQL Server でステートメントベースの [バイナリログ形式](#)を使用する場合 (デフォルトの動作です)、データベース内に一時テーブルがあるときにバックアップを取得し、これらの一時テーブルを使用して通常のテーブルへの更新または挿入を行うと、バックアップへの MySQL バイナリログの適用が失敗することがあります。つまり、MySQL バイナリログを使用して特定の時点にバックアップをロールフォワードできない可能性があります。これは、物理ファイル名 `#sql*.frm` が MySQL によりバイナリログに書き込まれる論理テーブル名に一致しないために、一時テーブルがバックアップにコピーされないからです。この問題を回避するには、サーバー上で `--binlog-format` オプションの値を「ROW」または「MIXED」に設定して、バイナリログに行ベースまたは混在形式を使用してください。

- `mysql.backup_history` テーブルの `engines` カラムには、バックアップされたデータベースのストレージエンジンは正しく反映されません。
- データディレクトリにテーブルを保存するときに、MySQL Server は、あらゆるデータベース名またはテーブル名内のすべての特殊文字に対して変換を実行し、ストレージ用のファイル名を生成します。次に、データベース名およびファイル名に特殊文字 (「`.`」と「`-`」) を含んだ MySQL コマンドの 2 つの例を示します。

```
mysql> CREATE TABLE `db.2`.t2 (c1 INT) ENGINE=InnoDB;
mysql> CREATE TABLE `db-3`.t-3 (c1 INT) ENGINE=InnoDB;
```

これらのコマンドで作成されたテーブルは、次のファイルとしてファイルシステムに格納されます。

```
db@002e2/t@002e2.ibd
db@002d3/t@002d3.ibd
```

現在、`mysqlbackup` は、`--include-tables` オプションと `--exclude-tables` オプションを使用して与えられたデータベース名およびテーブル名で、同じ変換を実行しません。たとえば、`--include-tables="db\2\t2|db-3\t-3"` を使用すると、前述のテーブルは選択されません。

回避策として、ユーザーは、データベースのデータディレクトリで目的のテーブルのファイル名を調べ、`--include-tables` オプションと `--exclude-tables` オプションの引数で、データベース名としてデータベースディレクトリの変換後の名前を使用し、テーブル名として変換後のファイル名 (ファイル拡張子なし) を使用することができます。たとえば、前述のテーブルの場合、オプション引数 `--include-tables="db@002e2\t@002e2|db@002d3\t@002d3"` で、バックアップに含めるテーブルとして選択します。オプション `--exclude-tables` で同じ引数を使用すると、これら 2 つのテーブルはバックアップで除外されます。

- 大量の書き込みワークロード (1 分あたり数ギガバイト程度) を抱える大規模データベースのホットバックアップは、バックアップの進行中にサーバーで生成される大きな Redo ログファイルのために、完了までに長時間かかることがあります。また、Redo ログファイルが `mysqlbackup` で可能な処理速度よりも速く増大した場合、`mysqlbackup` が Redo ログサイクルに追いつけず、LSN が `mysqlbackup` によって読み取られる前にサーバーによって上書きされたときに、バックアップ操作は実際に失敗することがあります。Redo ログの読み取りと書き込みに利用できる I/O リソースが、バックアッププロセス中に不足しているときに、問題は大きくなります。ただし、データベースの少数のテーブルだけが頻繁に変更される場合は、オペティミスティックバックアップ機能で問題が軽減される可能性があります。詳細は、[セクション 3.3.6 「オペティミスティックバックアップの作成」](#) を参照してください。
- MySQL Server 5.6.10 以前からの圧縮 InnoDB テーブルは、InnoDB ストレージエンジンでのバグのため、MySQL Enterprise Backup 3.9.0 以降ではリストアできません (MySQL Bug System で Bug# 72851 を参照してください)。
- サーバーからバックアップへバイナリログファイルをコピーしようとする (3.11.0 以降の MySQL Enterprise Backup のデフォルトの動作です)、増分バックアップの基準である完全バックアップが `--no-locking` オプションで作成されていた場合は、その増分バックアップは失敗します。回避策として、ユーザーは、この状況で実行される増分バックアップに `--skip-binlog` オプションを使用する必要があります。
- サーバーからバックアップへバイナリログファイルをコピーしようとする (3.11.0 以降の MySQL Enterprise Backup のデフォルトの動作です)、オフラインバックアップは失敗します。回避策として、オフラインバックアップには常に、`--skip-binlog` オプションを使用してください。
- MySQL Enterprise Backup を使用して、ネットワークアタッチトストレージ (NAS) デバイスへのバックアップ、またはそこからのリストアを行うことは可能ですが、ネットワーク問題が発生した場合には、バックアップの一貫性とバックアップまたはリストア操作のパフォーマンスが損なわれる恐れがあります。

付録 B MySQL Enterprise Backup の互換性情報

目次

B.1 プラットフォーム間の互換性	123
B.2 古い MySQL Enterprise Backup との互換性	123
B.3 古い MySQL バージョンとの互換性	123
B.4 特定の MySQL バージョンに関する互換性の注意事項	123

このセクションでは、MySQL Enterprise Backup リリースの互換性の問題に関連した情報について説明します。

B.1 プラットフォーム間の互換性

MySQL Enterprise Backup は、Linux および Windows オペレーティングシステムで実行する場合、プラットフォーム間で互換性があります。つまり、Linux マシン上でのバックアップは Windows マシン上でリストアでき、その逆も可能です。ただし、データベース名やテーブル名の太文字と小文字の相違から生じるデータ転送問題を回避するには、MySQL Server で変数 `lower_case_table_names` を適切に構成する必要があります。詳細は、[識別子の太文字と小文字の区別](#)を参照してください。

B.2 古い MySQL Enterprise Backup との互換性

MySQL Enterprise Backup 3.11 は、MySQL Enterprise Backup 3.9、3.10、および 3.11 で作成されたバックアップのリストアに使用できます。

B.3 古い MySQL バージョンとの互換性

MySQL データとログファイルの形式に変更が行われることがあります。これらの変更は、MySQL Server のバックアップを作成する方法に影響し、その結果、古い MySQL Enterprise Backup バージョンと新しい MySQL Server バージョンとの互換性、またはその逆の互換性の問題が生じる可能性があります。

現在、MySQL バージョン 5.5 および 5.6 との互換性は、MySQL Enterprise Backup 3.10 および 3.11 で保持されています。古いサーバーバージョンから新しいバージョンに (またはその逆に) データをリストアする場合、特別な計画と手順が必要になることに注意してください。詳細は、[MySQL のアップグレードとダウングレード](#)を参照してください。

B.4 特定の MySQL バージョンに関する互換性の注意事項

このセクションでは、バックアッププロセスのさまざまな側面に影響する、特定の MySQL Server バージョンでのパフォーマンス関連の機能および設定を一覧表示します。

MySQL 5.6

MySQL 5.6 の一部の新しい機能では、InnoDB テーブルのディレクトリレイアウトおよびファイルの内容が変更されています。次の機能を使用するサーバーをバックアップするには、MySQL Enterprise Backup 3.8.1 以上が必要です。

- `innodb_page_size` 構成オプション。
- `innodb_undo_directory`、`innodb_undo_logs`、および `innodb_undo_tablespaces` 構成オプション。
- `innodb_checksum_algorithm` 構成オプション。
- `CREATE TABLE` ステートメントの `DATA DIRECTORY` 句。これは、データベースディレクトリ内に `.isl` ファイルを作成し、ユーザー指定の場所に `.ibd` ファイルを格納します。
- [オンライン DDL](#)。

これらの MySQL 5.6 機能に関連した修正および拡張の詳細は、[MySQL Enterprise Backup 3.9 のリリースノート](#)を参照してください。

付録 C 詳細な例

目次

C.1 完全バックアップのディレクトリ構造例	125
C.2 圧縮バックアップのディレクトリ構造例	129
C.3 増分バックアップのディレクトリ構造例	129

このセクションでは、さまざまなバックアップ操作とリストア操作のコマンドおよび関連した出力について説明します。

C.1 完全バックアップのディレクトリ構造例

通常のバックアップディレクトリの下にあるサブディレクトリとファイルの例を次に示します。--with-timestamp オプションは、各バックアップの新しいサブディレクトリを作成し、ジョブのタイムスタンプに応じて名前を付けます。バックアップには、InnoDB システムテーブルスペースのファイル、さまざまなストレージエンジンからのテーブルおよびインデックスデータを表す .ibd、.frm、.MYD、.MYI、.CSV、および .CSM ファイル、パーティション化されたテーブルを表す .par および #P# ファイルが含まれます。

```
$ find ~/backups
/Users/cirrus/backups
/Users/cirrus/backups/2011-06-16_10-33-47
/Users/cirrus/backups/2011-06-16_10-33-47/backup-my.cnf
/Users/cirrus/backups/2011-06-16_10-33-47/datadir
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/ib_logfile0
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/ib_logfile1
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/ibbackup_logfile
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/ibdata1
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/db.opt
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p0.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p1.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p2.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p3.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p4.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p5.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p6.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p7.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double#P#p8.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_double.par
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p0.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p1.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p2.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p3.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p4.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p5.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p6.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p7.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long#P#p8.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_long.par
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p0.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p1.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p2.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p3.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p4.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p5.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p6.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p7.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string#P#p8.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/dc_p_string.par
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/graph_dc_schedules.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/graph_dc_schedules.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/graph_schedules.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/graph_schedules.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/graph_series_v2.frm
```


/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_alarms.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_dc_schedules.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_dc_schedules.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_eval_result_vars.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_eval_result_vars.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_eval_results.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_eval_results.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_schedule_email_targets.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_schedule_email_targets.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_schedules.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_schedules.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_tags.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_tags.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_thresholds.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_thresholds.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_variables.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rule_variables.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rules.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/rules.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/schema_version_v2.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/schema_version_v2.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_data.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_data.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_examples.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_examples.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_explain_data.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_explain_data.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_summaries.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_summaries.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_summary_data.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/statement_summary_data.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/system_maps.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/system_maps.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/tags.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/tags.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/target_email.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/target_email.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_form_defaults.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_form_defaults.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_preferences.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_preferences.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_tags.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/user_tags.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/users.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/users.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/whats_new_entries.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mem/whats_new_entries.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_history.CSM
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_history.CSV
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_history.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_progress.CSM
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_progress.CSV
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/backup_progress.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/columns_priv.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/columns_priv.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/columns_priv.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/db.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/db.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/db.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/event.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/event.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/event.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/func.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/func.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/func.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/general_log.CSM
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/general_log.CSV
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/general_log.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_category.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_category.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_category.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_keyword.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_keyword.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_keyword.MYI

```
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_relation.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_relation.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_relation.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_topic.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_topic.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/help_topic.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/host.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/host.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/host.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/ibbackup_binlog_marker.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/ibbackup_binlog_marker.ibd
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/inventory.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/inventory.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/inventory.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/ndb_binlog_index.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/ndb_binlog_index.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/ndb_binlog_index.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/plugin.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/plugin.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/plugin.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/proc.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/proc.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/proc.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/procs_priv.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/procs_priv.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/procs_priv.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/servers.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/servers.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/servers.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/slow_log.CSM
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/slow_log.CSV
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/slow_log.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/tables_priv.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/tables_priv.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/tables_priv.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_leap_second.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_leap_second.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_leap_second.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_name.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_name.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_name.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition_type.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition_type.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/time_zone_transition_type.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/user.frm
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/user.MYD
/Users/cirrus/backups/2011-06-16_10-33-47/datadir/mysql/user.MYI
/Users/cirrus/backups/2011-06-16_10-33-47/meta
/Users/cirrus/backups/2011-06-16_10-33-47/meta/backup_content.xml
/Users/cirrus/backups/2011-06-16_10-33-47/meta/backup_create.xml
/Users/cirrus/backups/2011-06-16_10-33-47/meta/backup_variables.txt
/Users/cirrus/backups/2011-06-16_10-34-12
/Users/cirrus/backups/2011-06-16_10-34-12/backup-my.cnf
/Users/cirrus/backups/2011-06-16_10-34-12/datadir
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/ib_logfile0
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/ib_logfile1
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/ibbackup_logfile
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/ibdata1
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/mem
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/mem/db.opt
/Users/cirrus/backups/2011-06-16_10-34-12/datadir/mem/dc_p_double#P#p0.ibd
...same database and table files as the previous backup...
/Users/cirrus/backups/2011-06-16_10-34-12/meta
/Users/cirrus/backups/2011-06-16_10-34-12/meta/backup_content.xml
/Users/cirrus/backups/2011-06-16_10-34-12/meta/backup_create.xml
/Users/cirrus/backups/2011-06-16_10-34-12/meta/backup_variables.txt
```


C.2 圧縮バックアップのディレクトリ構造例

MySQL Enterprise Monitor リポジトリデータベースからのバックアップの `backup-dir/datadir/mem` の下にあるファイルリストの抜粋を次に示します。現在、InnoDB テーブルの `.ibd` ファイルは `.ibz` ファイルに圧縮されますが、ほかの種類のファイルは変更されていないことに注意してください。

```
inventory_types.frm
inventory_types.ibz
log_db_actions#P#p0.ibz
log_db_actions#P#p1.ibz
log_db_actions#P#p2.ibz
log_db_actions#P#p3.ibz
log_db_actions#P#p4.ibz
log_db_actions#P#p5.ibz
log_db_actions#P#p6.ibz
log_db_actions#P#p7.ibz
log_db_actions#P#p8.ibz
log_db_actions.frm
log_db_actions.par
loghistogram_data.frm
loghistogram_data.ibz
```

C.3 増分バックアップのディレクトリ構造例

増分バックアップは、完全バックアップからのファイルのサブセットを含むディレクトリ構造を作成します。`*.frm` `*.MYD` などの非 InnoDB ファイルはすべて含まれます。`*.ibd` ファイルは、完全バックアップ以降に変更があった場合にのみ、つまり、最大の論理シーケンス番号が `--start-lsn` オプションで指定された値よりも高くなった場合にのみ含まれます。

```
$ find /tmp/backups
/tmp/backups
/tmp/backups/backup-my.cnf
/tmp/backups/datadir
/tmp/backups/datadir/ibbackup_ibd_files
/tmp/backups/datadir/ibbackup_logfile
/tmp/backups/datadir/ibdata1
/tmp/backups/datadir/mem
/tmp/backups/datadir/mem/db.opt
/tmp/backups/datadir/mem/dc_p_double.frm
/tmp/backups/datadir/mem/dc_p_double.par
/tmp/backups/datadir/mem/dc_p_long.frm
/tmp/backups/datadir/mem/dc_p_long.par
/tmp/backups/datadir/mem/dc_p_string.frm
/tmp/backups/datadir/mem/dc_p_string.par
/tmp/backups/datadir/mem/graph_dc_schedules.frm
/tmp/backups/datadir/mem/graph_schedules.frm
... many more files...
```

付録 D MySQL Enterprise Backup リリースノート

MySQL Enterprise Backup のリリースノートは個別に発行されます。[MySQL Enterprise Backup 3.11 リリースノート](#)を参照してください。

付録 E サードパーティーコンポーネントライセンス

目次

E.1 RegEX-Spencer ライブラリライセンス	133
E.2 zlib ライセンス	133
E.3 LZ4 ライセンス	134
E.4 Percona Multiple I/O スレッドパッチライセンス	134
E.5 Google SMP Patch ライセンス	135
E.6 Google Controlling Master Thread I/O Rate Patch ライセンス	135
E.7 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) ライセンス	136
E.8 OpenSSL v1.0 ライセンス	136
E.9 cURL (libcurl) ライセンス	138

Oracle は、特定のサードパーティーおよびオープンソースソフトウェアが MySQL Enterprise Backup 製品の開発に使用されている、または製品に組み込まれていることを認識しています。この付録には、必要なサードパーティーのライセンス情報が含まれます。

E.1 RegEX-Spencer ライブラリライセンス

この製品には、次のソフトウェアが含まれている場合があります。

```
Henry Spencer's Regular-Expression Library (RegEX-Spencer)

Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone
and Telegraph Company or of the Regents of the University of
California.

Permission is granted to anyone to use this software for any purpose
on any computer system, and to alter it and redistribute it, subject
to the following restrictions:

1. The author is not responsible for the consequences of use of this
software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by
explicit claim or by omission. Since few users ever read sources,
credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be
misrepresented as being the original software. Since few users ever
read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.
```

E.2 zlib ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

zlib

Oracle は、この製品で使用される zlib 汎用圧縮ライブラリの作成における Jean-loup Gailly 氏と Mark Adler 氏の貢献に深く感謝します。

```
zlib.h -- interface of the 'zlib' general purpose compression library
Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
```

version 1.2.5, April 19th, 2010
 Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
 Mark Adler madler@alummi.caltech.edu

E.3 LZ4 ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

LZ4 - Fast LZ compression algorithm

Copyright (C) 2011-2013, Yann Collet.
 BSD 2-Clause License (<http://www.opensource.org/licenses/bsd-license.php>)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
 - LZ4 source repository : <http://code.google.com/p/lz4/>
 - LZ4 public forum : <https://groups.google.com/forum/#!forum/lz4c>

E.4 Percona Multiple I/O スレッドパッチライセンス

この製品には、次のソフトウェアが含まれている場合があります。

Percona Multiple I/O スレッドパッチ

Copyright (c) 2008, 2009 Percona Inc
 All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Percona Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of Percona Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

E.5 Google SMP Patch ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

Google SMP Patch

Google SMP patch

Copyright (c) 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

E.6 Google Controlling Master Thread I/O Rate Patch ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

Google Controlling Master Thread I/O Rate Patch

Copyright (c) 2009, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

E.7 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement
 Funding for the RFC Editor function is currently provided by the Internet Society.

E.8 OpenSSL v1.0 ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

OpenSSL v1.0

NOTE: Does not apply to GPL licensed server (OpenSSL is not shipped with it)

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of

the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

/ =====

Copyright (c) 1998-2008 The OpenSSL Project.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (Link1 /)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (Link2 /)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

/ Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic

related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The license and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

E.9 cURL (libcurl) ライセンス

この製品には、次のソフトウェアが含まれている場合があります。

cURL (libcurl)

Use of any of this software is governed by the terms of the license below:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2014, Daniel Stenberg, <daniel@haxx.se>.
All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

MySQL Enterprise Backup 用語集

これらの用語は、MySQL Enterprise Backup 製品に関する情報で一般的に使用されます。

.

.ARM ファイル

ARCHIVE テーブルのメタデータ。 .ARZ ファイルと対比してください。この拡張子を持つファイルは常に、MySQL Enterprise Backup 製品の `mysqlbackup` コマンドで生成されたバックアップに含まれます。

[.ARZ ファイル](#), [MySQL Enterprise Backup](#)も参照

.ARZ ファイル

ARCHIVE テーブルのデータ。 .ARM ファイルと対比してください。この拡張子を持つファイルは常に、MySQL Enterprise Backup 製品の `mysqlbackup` コマンドで生成されたバックアップに含まれます。

[.ARM ファイル](#), [MySQL Enterprise Backup](#)も参照

.frm ファイル

MySQL テーブルのメタデータ (テーブル定義など) を含むファイル。

バックアップの場合、バックアップ後に変更または削除されたテーブルをリストアできるように、バックアップデータとともに `.frm` ファイルの完全セットを常に保持する必要があります。

それぞれの InnoDB テーブルには `.frm` ファイルがありますが、InnoDB は独自のテーブルメタデータをシステムテーブルスペースに保持しています。InnoDB が InnoDB テーブルを処理する場合、 `.frm` ファイルは不要です。

これらのファイルは、MySQL Enterprise Backup 製品でバックアップされます。バックアップが行われている間にこれらのファイルを `ALTER TABLE` 操作で変更してはいけないため、InnoDB でないテーブルを含むバックアップは `.frm` ファイルのバックアップ中に、`FLUSH TABLES WITH READ LOCK` 操作を実行してこのようなアクティビティーをフリーズします。バックアップをリストアするときに、バックアップ時点のデータベースの状態に一致するように、 `.frm` ファイルが作成、変更、または削除される場合があります。

[MySQL Enterprise Backup](#)も参照

.ibd ファイル

file-per-table 設定を使用して作成された各 InnoDB テーブルスペースには、 `.ibd` の拡張子を含むファイル名が付けられます。この拡張子は、 `ibdata1` や `ibdata2` などの名前が付けられたファイルから構成されるシステムテーブルスペースには適用されません。

[.ibd ファイル](#), [システムテーブルスペース](#), [テーブルスペース](#)も参照

.ibz ファイル

MySQL Enterprise Backup 製品は圧縮バックアップを実行するときに、file-per-table 設定を使用して作成される各テーブルスペースファイルを、 `.ibd` 拡張子から `.ibz` 拡張子に変換します。

バックアップ中に適用される圧縮は、通常操作中にテーブルデータを圧縮されたままにする圧縮行フォーマットとは異なります。すでに圧縮行フォーマットになっている InnoDB テーブルスペースは、これ以上圧縮しても領域はほとんどまたはまったく節約されないため、2 度目の圧縮は行われません。

[.ibd ファイル](#), [.ibz ファイル](#), [MySQL Enterprise Backup](#), [テーブルスペース](#)も参照

.MRG ファイル

`MERGE` ストレージエンジンで使用される、ほかのテーブルへの参照を含むファイル。この拡張子を持つファイルは常に、MySQL Enterprise Backup 製品の `mysqlbackup` コマンドで生成されたバックアップに含まれます。

[MySQL Enterprise Backup](#)も参照

.MYD ファイル

MyISAM テーブルのデータを格納するために MySQL が使用するファイル。

[.MYI ファイル](#), [MySQL Enterprise Backup](#)も参照

.MYI ファイル

MyISAM テーブルのインデックスを格納するために MySQL が使用するファイル。

[.MYD ファイル](#), [MySQL Enterprise Backup](#)も参照

.opt ファイル

データベース構成情報を含むファイル。この拡張子のファイルは常に、MySQL Enterprise Backup 製品のバックアップ操作で生成されたバックアップに含まれます。

[MySQL Enterprise Backup](#)も参照

.par ファイル

パーティション定義を含むファイル。この拡張子を持つファイルは常に、MySQL Enterprise Backup 製品の `mysqlbackup` コマンドで生成されたバックアップに含まれます。

[MySQL Enterprise Backup](#)も参照

.TRG ファイル

トリガーパラメータを含むファイル。この拡張子を持つファイルは常に、MySQL Enterprise Backup 製品の `mysqlbackup` コマンドで生成されたバックアップに含まれます。

[MySQL Enterprise Backup](#)も参照

A

Antelope

元の InnoDB ファイル形式のコード名。これは、冗長およびコンパクトの行フォーマットをサポートしますが、Barracuda ファイル形式で利用できる新しい動的および圧縮の行フォーマットはサポートしません。

アプリケーションが、InnoDB テーブル圧縮からメリットを受けられるか、動的行フォーマットからメリットを受けられる BLOB またはラージテキストカラムを使用する場合、一部のテーブルを Barracuda 形式に切り替えることができます。テーブルを作成する前に、`innodb_file_format` オプションを設定することによって、使用するファイル形式を選択します。

[Barracuda](#), [圧縮](#), [ファイル形式](#)も参照

B

backup-my.cnf

構成パラメータの最低限のセットを含む、MySQL Enterprise Backup によって生成された小さな構成ファイル。このファイルは、このバックアップデータに適用される設定を記録します。適用プロセスなどの後続の操作で、このファイルからオプションを読み取って、バックアップデータの構造を判断します。このファイルには常に、Unix のようなシステムでの `.cnf` や Windows システムでの `.ini` ではなく、`.cnf` の拡張子が付けられます。

[適用](#), [構成ファイル](#)も参照

Barracuda

テーブルデータの圧縮をサポートする InnoDB ファイル形式のコード名。このファイル形式は、最初に InnoDB Plugin に導入されました。これは、InnoDB テーブル圧縮に有効にする圧縮行フォーマットと、BLOB およびラージテキストカラムのストレージレイアウトを改善する動的行フォーマットをサポートします。`innodb_file_format` オプションを通じてこれを選択できます。

InnoDB システムテーブルスペースは元の Antelope ファイル形式で格納されるため、Barracuda ファイル形式を使用するには、システムテーブルスペースとは別の独自のテーブルスペースに新しく作成したテーブルを格納する `file-per-table` 設定も有効にする必要があります。

MySQL Enterprise Backup 製品バージョン 3.5 以上は、Barracuda ファイル形式を使用するテーブルスペースのバックアップをサポートします。

[Antelope](#), [ファイル形式](#), [MySQL Enterprise Backup](#), [行フォーマット](#), [システムテーブルスペース](#)も参照

binlog

バイナリログファイルの非公式名。たとえば、電子メールメッセージやフォーラムディスカッションでこの略語を見ることがあります。

[バイナリログ](#)も参照

I

ibdata ファイル

InnoDB システムテーブルスペースを構成する、`ibdata1` や `ibdata2` などの名前を持つファイルのセット。これらのファイルには、InnoDB テーブルに関するメタデータが含まれ、テーブルおよびインデックスデータの一部またはすべても含まれることがあります (各テーブルが作成されるときに、`file-per-table` オプションが有効であるかどうかによって異なります)。後方互換性のため、これらのファイルは常に、Antelope ファイル形式を使用します。

[Antelope](#), [システムテーブルスペース](#)も参照

InnoDB

MySQL Enterprise Backup でもっとも効果的に機能する MySQL テーブルのタイプ。これらのテーブルは、データベース処理の一時停止を回避するホットバックアップの手法を使用してバックアップできます。この理由と、InnoDB テーブルで得られるより高い信頼性と並列性のため、ほとんどの配備での大部分のデータと最も重要なデータには、InnoDB を使用してください。MySQL 5.5 以上では、[CREATE TABLE](#) ステートメントはデフォルトで InnoDB テーブルを作成します。[ホットバックアップ](#), [MySQL Enterprise Backup](#), [テーブル](#)も参照

L

LSN

ログシーケンス番号の頭字語。この任意の増加し続ける値は、Redo ログに記録された操作に応じた特定の時点を表します。(この特定の時点は、トランザクション限界には関係ありません。1 つ以上のトランザクションの最中にあたることもあります。)クラッシュリカバリ中に InnoDB によって内部的に、バッファプールを管理するために使用されます。

MySQL Enterprise Backup 製品では、増分バックアップを取得する時点を表す LSN を指定できます。該当する LSN は、[mysqlbackup](#) コマンドの出力で表示されます。完全バックアップの時点に対応する LSN がわかれば、後続の増分バックアップを取得するためにその値を指定でき、その出力には次の増分バックアップのもう 1 つの LSN が含まれます。[クラッシュリカバリ](#), [ホットバックアップ](#), [増分バックアップ](#), [Redo ログ](#)も参照

M

my.cnf

Linux、Unix、および OS X システムでの MySQL 構成ファイルの標準的な名前。[構成ファイル](#), [my.ini](#)も参照

my.ini

Windows システムでの MySQL 構成ファイルの標準的な名前。[構成ファイル](#), [my.cnf](#)も参照

MyISAM

以前は新しいテーブルのデフォルトであった MySQL ストレージエンジン。MySQL 5.5 以上では、InnoDB がデフォルトのストレージエンジンになっています。MySQL Enterprise Backup は、両方のタイプのテーブルと、ほかのストレージエンジンからのテーブルもバックアップできます。InnoDB テーブルのバックアッププロセス (ホットバックアップ) は、MyISAM テーブルのバックアッププロセス (ウォームバックアップ) よりもデータベース操作を妨げることが少なくなります。[ホットバックアップ](#), [InnoDB](#), [MySQL Enterprise Backup](#), [ウォームバックアップ](#)も参照

MySQL Enterprise Backup

MySQL データベースのホットバックアップを実行するライセンス製品。InnoDB テーブルをバックアップするときにもっとも高い効率性と柔軟性をもたらします。MyISAM とほかの種類のテーブルをバックアップすることもできます。MySQL Enterprise Edition サブスクリプションの一部として含まれています。[Barracuda](#), [ホットバックアップ](#), [InnoDB](#)も参照

mysqlbackup

MySQL Enterprise Backup 製品の主要なコマンド。さまざまなオプションがバックアップ操作とリストア操作を実行します。[バックアップ](#), [MySQL Enterprise Backup](#), [リストア](#)も参照

mysqldump

論理バックアップを実行する MySQL コマンドであり、テーブルおよびデータを再度作成する一連の SQL コマンドを生成します。MySQL Enterprise Backup で生成される物理バックアップの場合よりもリストア操作に時間がかかるため、小さなバックアップや重要度の低いデータに適しています。[論理バックアップ](#), [MySQL Enterprise Backup](#), [物理バックアップ](#), [リストア](#)も参照

O

Oracle Secure Backup

バックアップメディアを管理するための Oracle 製品。メディア管理ソフトウェア (MMS) として分類されます。OSB と略記されます。MySQL Enterprise Backup の場合、OSB は通常、テープバックアップの管理に使用されます。[バックアップ](#), [メディア管理ソフトウェア](#), [MySQL Enterprise Backup](#), [OSB](#)も参照

OSB

Oracle Secure Backup の略語であり、メディア管理ソフトウェア製品 (MMS) です。
[Oracle Secure Backup](#)も参照

R

raw バックアップ

バックアップの実行中に行われた変更を組み込んでいないため、まだリストアできる準備が整っていないバックアップデータの初期セット。適用操作は、バックアップファイルを、リストアできる準備の整った準備されたバックアップに変換します。

[適用, 準備されたバックアップ](#)も参照

Redo ログ

通常は `ib_logfile0` および `ib_logfile1` の名前が付けられた一連のファイルであり、InnoDB テーブルのデータを変更しようとするステートメントを記録します。クラッシュ後の起動時に、これらのステートメントは自動的に再実行され、不完全なトランザクションによって書き込まれたデータを修正します。Redo ログでのデータの推移は、増加し続ける LSN 値によって表されます。MySQL 5.6 では、Redo ログのサイズの上限が 4G バイトより大きくなっています。

[LSN](#)も参照

S

SBT

System Backup to Tape (テープへのバックアップ) の頭字語。

[System Backup to Tape](#)も参照

System Backup to Tape

メディア管理ソフトウェアの API。SBT と略記されます。いくつかの (名前に `sbt` を含む) `mysqlbackup` オプションは、Oracle Secure Backup などのメディア管理ソフトウェア製品に情報を渡します。

[Oracle Secure Backup, SBT](#)も参照

ア

圧縮

より小さなバックアップファイルを生成する手法で、サイズの縮小は圧縮レベル設定に影響されます。重要でない複数セットのバックアップファイルを保持する場合に最適です。(重大なデータの最新のバックアップの場合、データを圧縮せずに置いておくと、緊急時に高速でリストアすることが可能になります。)

MySQL Enterprise Backup は、バックアッププロセス中に InnoDB テーブルの内容に圧縮を適用でき、`.ibd` ファイルを `.ibz` ファイルに変換します。

圧縮によって、バックアッププロセスでの CPU オーバーヘッドが増大し、リストアプロセス中に必要になる時間とディスク領域が増加します。

[バックアップ, 圧縮レベル, .ibd ファイル, .ibz ファイル, InnoDB, MySQL Enterprise Backup, リストア](#)も参照

圧縮レベル

圧縮バックアップに適用される圧縮の程度を決定する設定。この設定の範囲は、0 (なし) から、1 (圧縮が有効なときのデフォルトレベル)、9 (最大) までです。所定の圧縮レベルの圧縮の量は、データ値の性質に応じて異なります。圧縮レベルを高めると CPU オーバーヘッドが増大することになるため、低い CPU オーバーヘッドで、良好なバランスの圧力を生成する最低の値を使用することが理想的です。

[圧縮](#)も参照

イ

イメージ

単一ファイルバックアップ操作の一部として生成されるファイル。これはローカルで格納する実際のファイルの場合も、バックアップデータが別のコマンドまたはリモートサーバーに直接ストリーミングされるとき (-として指定される) 標準出力の場合もあります。この用語は、`backup-dir-to-image` や `image-to-backup-dir` などの複数の `mysqlbackup` オプションで参照されます。

[単一ファイルバックアップ, ストリーミング](#)も参照

インスタンス

MySQL Server の完全な内容。複数のデータベースが含まれる可能性があります。バックアップ操作でインスタンス全体をバックアップすることも、部分バックアップで選択したデータベースとテーブルを含めることもできます。

[データベース](#), [部分バックアップ](#)も参照

一時停止

ユーザー固有の操作を実行できるようにするために、MySQL Enterprise Backup の処理が停止するバックアップ内のオプションステージ。mysqlbackup コマンドには、バックアップが一時停止されている間に行うコマンドを指定するためのオプションがあります。ほとんどの場合、.frm ファイルを処理するための自身のスクリプト処理を実行できる InnoDB テーブルのバックアップとともにのみ使用されます。

[.frm ファイル](#), [InnoDB](#)も参照

ウ

ウォームバックアップ

データベースの実行中に行われるが、バックアッププロセス中に一部のデータベース操作を制限するバックアップ。たとえば、テーブルが読み取り専用になることがあります。ビジーなアプリケーションおよび Web サイトの場合、ホットバックアップをお勧めします。

[バックアップ](#), [コールドバックアップ](#), [ホットバックアップ](#)も参照

オ

オフライン

データベースサーバーが停止している間に行われる操作の種類。MySQL Enterprise Backup 製品では、主要なオフライン操作はリストアステップです。オプションで、別のオフライン操作であるコールドバックアップを行うことができます。「オンライン」と対比してください。

[コールドバックアップ](#), [MySQL Enterprise Backup](#), [オンライン](#), [リストア](#)も参照

オンライン

データベースサーバーが実行している間に行われる操作の種類。ホットバックアップが理想的な例です。データベースを継続して実行でき、読み取り操作や書き込み操作がブロックされないためです。この理由により、「ホットバックアップ」と「オンラインバックアップ」はシノニムとして使用されることがあります。コールドバックアップは、オンライン操作の反対です。定義上、このバックアップ中は、データベースサーバーはシャットダウンしています。データベースサーバーが実行し続けるため、ウォームバックアップもオンライン操作の一種です。ただし、ウォームバックアップの進行中、一部の書き込み操作はブロックされることがあります。「オフライン」と対比してください。

[コールドバックアップ](#), [ホットバックアップ](#), [オフライン](#), [ウォームバックアップ](#)も参照

カ

完全バックアップ

各 MySQL データベース内のすべてのテーブルと、MySQL インスタンス内のすべてのデータベースを含むバックアップ。「部分バックアップ」および「増分バックアップ」と対比してください。完全バックアップはもっとも時間がかかりますが、必要になるフォローアップ作業量と管理上の複雑さはもっとも少なくなります。したがって、部分バックアップまたは増分バックアップを主に行う場合でも、定期的に完全バックアップを行うことができます。

[バックアップ](#), [増分バックアップ](#), [部分バックアップ](#), [テーブル](#)も参照

ク

クラッシュリカバリ

クラッシュ後に MySQL を再度起動したときに行われる、InnoDB テーブルのクリーンアップアクティビティ。クラッシュ前にコミットされたが、まだテーブルスペースファイルに書き込まれていない変更は、二重書き込みバッファから再構築されます。通常どおりデータベースがシャットダウンした場合、このタイプのアクティビティは、ページ操作によってシャットダウン中に実行されます。

コ

コールドバックアップ

データベースがシャットダウンしている間に行われるバックアップ。ビジーなアプリケーションおよび Web サイトの場合、これは実用的でない可能性があり、ウォームバックアップまたはホットバックアップをお勧めします。

[バックアップ](#), [接続](#), [ホットバックアップ](#), [ウォームバックアップ](#)も参照

構成ファイル

MySQL サーバーと関連製品およびコンポーネントのスタートアップオプションを保持するファイル。多くの場合、Linux、Unix、および OS X システムでは my.cnf、Windows システムでは my.ini という、そのデフォルトのファイル名で呼ばれます。MySQL Enterprise Backup はそのデフォルトの構成設定をこのファイルの「mysqlbackup」セクションの下に格納します。MySQL Enterprise Backup は便宜上、MySQL Enterprise Backup と、MySQL Server に接続するほかのプログラムとの間で共通した構成オプションについての設定を、「client」セクションから読み取ることもできます。
[my.cnf](#), [my.ini](#), [MySQL Enterprise Backup](#)も参照

サ

サーバー

mysqld デーモンで制御される MySQL インスタンス。1 台の物理マシンで複数の MySQL Server をホストでき、それぞれに専用のバックアップ操作とスケジュールが必要です。一部のバックアップ操作は、接続を通じてサーバーと通信します。
[接続](#), [インスタンス](#)も参照

サーバーリポジトリ

「バックアップリポジトリ」と対比してください。
[バックアップリポジトリ](#), [リポジトリ](#)も参照

シ

システムテーブルスペース

デフォルトでは、この単一データファイルには、データベースのすべてのテーブルデータと、InnoDB 関連のオブジェクトのすべてのメタデータ (データディクショナリ) が格納されます。

innodb_file_per_table オプションをオンにすると、新しく作成した各テーブルが専用のテーブルスペースに格納されるため、システムテーブルスペースのサイズと依存関係を軽減できます。

システムテーブルスペース内にすべてのテーブルデータを保持することは、(小さな複数のファイルではなく 1 つの大きなファイルをバックアップする) MySQL Enterprise Backup 製品に関連しており、新しい Barracuda ファイル形式を必要とする特定の InnoDB 機能を使用できないようにします。

[Barracuda](#), [データディクショナリ](#), [ファイル形式](#), [ibdata ファイル](#), [テーブルスペース](#)も参照

準備されたバックアップ

全体的に一貫性があり、リストアする準備の整っているバックアップデータのセット。これは、raw バックアップで適用操作を実行することによって作成されます。

[適用](#), [raw バックアップ](#)も参照

進行状況テーブル

実行中のバックアップ操作の詳細を保持するテーブル `mysql.backup_progress`。バックアップジョブが終了すると、詳細は履歴テーブルに記録されます。

[バックアップ](#), [履歴テーブル](#)も参照

除外

部分バックアップで、バックアップから省くテーブル、データベース、またはこれらの両方の組み合わせを選択すること。「含める」と対比してください。

[部分バックアップ](#)も参照

ス

ストリーミング

ローカルコピーを保存するのではなく、即座に別のサーバーにデータを転送するバックアップ手法。Unix パイプなどのメカニズムを使用します。バックアップ先ファイルが - (標準出力) として指定された単一ファイルバックアップが必要です。

[単一ファイルバックアップ](#)も参照

スレーブ

レプリケーション構成で、更新をマスターサーバーから受信するデータベースサーバーです。一般に、ユーザークエリーに対応する際に使用され、マスター上のクエリーの負荷を最小限にします。MySQL Enterprise Backup では、あるサーバー上

でバックアップを取得し、別のシステム上でリストアすることで、データが用意された新しいスレーブサーバーを作成できます。マスターではなくスレーブサーバーからデータをバックアップして、システム全体の速度低下を最低限に抑えることもできます。

[マスター](#)、[レプリケーション](#)も参照

セ

接続

実行中の MySQL Server と通信するために、特定のバックアップ操作で使用されるメカニズム。たとえば、`mysqlbackup` コマンドは、バックアップされているサーバーにログインして、進行状況テーブルと履歴テーブル内でデータを挿入および更新できます。通常ホットバックアップは、便宜上データベース接続を使用しますが、接続が使用できない場合でも進行することができます。ウォームバックアップは、サーバーを読み取り専用状態にする必要があるため、常にデータベース接続を使用します。コールドバックアップは MySQL Server のシャットダウン中に行われるため、接続が必要な機能はどれも使用できません。

[コールドバックアップ](#)、[履歴テーブル](#)、[ホットバックアップ](#)、[進行状況テーブル](#)、[サーバー](#)、[ウォームバックアップ](#)も参照

正規表現

MySQL Enterprise Backup の一部の機能では、たとえば、部分バックアップで含めるまたは除外するテーブルまたはデータベース、あるいはその両方を指定するために、POSIX スタイルの正規表現を使用します。正規表現では、ドットが単一文字のワイルドカードを表すため、ファイル名内のドットをエスケープする必要があります。パス名内のスラッシュのエスケープは不要です。コマンド行で正規表現を指定する場合は、シェルワイルドカードメカニズムでアスタリスクなどの文字に拡張されないようにするため、シェル環境の必要に応じて、引用符でこれらを囲みます。

[除外](#)、[含める](#)、[部分バックアップ](#)も参照

ソ

増分バックアップ

前回のバックアップ以降に変更のあったデータだけを取得するバックアップ。完全バックアップより小さく高速になる可能性があります。増分バックアップデータは、リストアする前に、前回のバックアップの内容とマージする必要があります。使用法の詳細は、[セクション3.3.2「増分バックアップの作成」](#)を参照してください。関連する `mysqlbackup` オプションは、`--incremental`、`--incremental-with-redo-log-only`、`--incremental-backup-dir`、`--incremental-base`、および `--start-lsn` です。

[完全バックアップ](#)も参照

タ

単一ファイルバックアップ

ストレージと転送を簡単にするために、すべてのバックアップデータを1つのファイル(バックアップイメージ)にバックアップするバックアップ手法。ストリーミングバックアップ手法では、単一ファイルバックアップを使用する必要があります。

[イメージ](#)、[ストリーミング](#)も参照

チ

抽出

単一ファイルバックアップによって生成されたイメージファイルから内容を取得する操作。これは、単一ファイルに適用することも(任意の場所にアンパックされます)、バックアップ全体に適用することも(バックアップデータの元のディレクトリ構造を再生成します)できます。これらの2種類の抽出は、`mysqlbackup` のオプションである `extract` と `image-to-backup-dir` をそれぞれ指定することによって実行されます。

[イメージ](#)、[単一ファイルバックアップ](#)も参照

テ

テーブル

SQL コンテキストのテーブルは個別のアドレス指定可能なオブジェクトですが、バックアップ目的では多くの場合、テーブルがシステムテーブルスペースの一部であるか、`file-per-table` 設定で作成されて独自のテーブルスペースに存在するかが重要です。

[バックアップ](#), [システムテーブルスペース](#), [テーブルスペース](#)も参照

テーブルスペース

InnoDB テーブルの場合、テーブルのデータとインデックスを保持するファイル。複数のテーブルを含むシステムテーブルスペースの場合も、独自のテーブルスペースファイルに存在する file-per-table 設定で作成されたテーブルの場合もあります。

[InnoDB](#), [システムテーブルスペース](#)も参照

データディクショナリ

テーブル、インデックス、テーブルカラムなどの InnoDB 関連のオブジェクトを追跡する、InnoDB ストレージエンジンで制御される一連のテーブル。これらのテーブルは、InnoDB システムテーブルスペースの一部です。

MySQL Enterprise Backup 製品は常にシステムテーブルスペースをバックアップするため、すべてのバックアップにデータディクショナリの内容が含まれます。

[ホットバックアップ](#), [MySQL Enterprise Backup](#), [システムテーブルスペース](#)も参照

データベース

MySQL ユーザーが所有する一連のテーブルと関連オブジェクト。Oracle Database 用語の「スキーマ」と同等です。MySQL Enterprise Backup は、一部のデータベースを含み、それ以外は含まない部分バックアップを実行できます。MySQL Server が制御するデータベースの完全なセットは、インスタンスと呼ばれます。

[インスタンス](#), [MySQL Enterprise Backup](#), [部分バックアップ](#)も参照

停止時間

データベースが応答しない期間。データベースは全体がシャットダウンしている場合も、アプリケーションがデータを挿入、更新、または削除しようとするときに読み取り専用状態になっている場合もあります。バックアップ戦略の目標は、InnoDB テーブル用のホットバックアップや、レプリケーション構成でのスレーブサーバーを使用したコールドバックアップなどの手法を使用し、MySQL Server がロックされている間にカスタマイズしたバックアップロジックを実行する一時停止ステージの期間を最小にして、停止時間を最低限に抑えることです。

[コールドバックアップ](#), [ホットバックアップ](#), [InnoDB](#), [ロック](#), [レプリケーション](#), [スレーブ](#), [一時停止](#)も参照

適用

ログからのデータを使用して、バックアップの実行中に行われた変更を組み込むことによって、raw バックアップを準備されたバックアップに変換する操作。

[ログ](#), [準備されたバックアップ](#), [raw バックアップ](#)も参照

ト

特定の時点

バックアップ操作の終了時にあたる時間。準備されたバックアップには、バックアップ操作の実行中に行われた変更がすべて含まれます。バックアップのリストアは、バックアップ操作が完了した時点の状態にデータを戻します。

[バックアップ](#), [準備されたバックアップ](#), [リストア](#)も参照

ハ

バイナリログ

テーブルデータを変更しようとするすべてのステートメントのレコードを含むファイル。レプリケーションシナリオでスレーブサーバーを最新にしたり、バックアップからテーブルデータをリストアしたあとでデータベースを最新にしたりするために、これらのステートメントを再現できます。バイナリロギング機能はオンとオフを切り替えられますが、レプリケーションを使用するかバックアップを実行する場合には、常にこれを有効にすることをお勧めします。

[mysqlbinlog](#) コマンドを使用することによって、バイナリログの内容を調べたり、レプリケーションまたはリカバリ中にこれらのステートメントを再現したりできます。バイナリログの詳細は、[バイナリログ](#)を参照してください。バイナリログに関連した MySQL 構成オプションについては、[バイナリログのオプションと変数](#)を参照してください。

MySQL Enterprise Backup 製品の場合、バイナリログのファイル名とファイル内での現在の位置が重要な詳細です。レプリケーションのコンテキストでバックアップを取得するときに、マスターサーバーに関するこの情報を記録するため `--slave-info` オプションを指定できます。

MySQL 5.0 より前では、更新ログと呼ばれる同様の機能を利用できました。MySQL 5.0 以上では、更新ログがバイナリログに置き換わりました。

[binlog](#), [MySQL Enterprise Backup](#), [レプリケーション](#)も参照

バックアップ

安全策として、MySQL インスタンスから一部またはすべてのテーブルデータおよびメタデータをコピーするプロセス。コピーされたファイルのセットを指す場合もあります。これは DBA のきわめて重要なタスクです。このプロセスの反対がリストア操作です。

MySQL では、物理バックアップは MySQL Enterprise Backup 製品で実行され、論理バックアップは `mysqldump` コマンドで実行されます。これらの方法は、バックアップデータのサイズおよび表現と速度 (特にリストア操作の速度) の点で特性が異なります。

バックアップはさらに、通常のデータベース操作に干渉する程度に応じて、ホット、ウォーム、コールドに分類されます。(干渉の程度は、ホットバックアップがもっとも少なく、コールドバックアップがもっとも多くなります。)

[コールドバックアップ](#)、[ホットバックアップ](#)、[論理バックアップ](#)、[MySQL Enterprise Backup](#)、[mysqldump](#)、[物理バックアップ](#)、[ウォームバックアップ](#)も参照

バックアップリポジトリ

「サーバーリポジトリ」と対比してください。

[リポジトリ](#)、[サーバーリポジトリ](#)も参照

フ

ファイル形式

`ibdata1` や `ibdata2` などの名前が付けられた InnoDB のデータファイルに対して、InnoDB で使用する形式。それぞれのファイル形式は、1 つ以上の行フォーマットをサポートします。

[Antelope](#)、[Barracuda](#)、[ibdata ファイル](#)、[行フォーマット](#)も参照

含める

部分バックアップで、バックアップするテーブル、データベース、またはこれらの両方の組み合わせを選択すること。「除外」と対比してください。

[部分バックアップ](#)も参照

物理バックアップ

実際のデータファイルをコピーするバックアップ。たとえば、MySQL Enterprise Backup のコマンドでは、その出力に `mysqld` サーバーで直接使用できるデータファイルが含まれているため、物理バックアップを作成します。論理バックアップと対比してください。

[バックアップ](#)、[論理バックアップ](#)、[MySQL Enterprise Backup](#)も参照

部分バックアップ

MySQL データベースのテーブルの一部、または MySQL インスタンスのデータベースの一部を含むバックアップ。完全バックアップと対比してください。

[バックアップ](#)、[完全バックアップ](#)、[部分リストア](#)、[テーブル](#)も参照

部分リストア

1 つ以上のテーブルまたはデータベースに適用されるが、MySQL Server の内容全体には適用されないリストア操作。リストアされるデータは、部分バックアップからの場合も、完全バックアップからの場合もあります。

[データベース](#)、[完全バックアップ](#)、[部分バックアップ](#)、[リストア](#)、[テーブル](#)も参照

へ

並列バックアップ

MySQL Enterprise Backup 3.8 以上でのデフォルトの処理モードであり、さまざまなクラスの内部操作 (読み取り、処理、および書き込み) に応じて複数のスレッドを使用します。概要については[セクション1.3「バックアップのパフォーマンスおよび容量に関する考慮事項の概要」](#)を、該当する `mysqlbackup` オプションについては[セクション5.1.11「パフォーマンス/スケラビリティ/容量オプション」](#)を、パフォーマンスのガイドラインとヒントについては[第7章「MySQL Enterprise Backup のパフォーマンスに関する考慮事項」](#)を参照してください。

ホ

ホットバックアップ

MySQL インスタンスが実行中で、アプリケーションがそれに対する読み取りおよび書き込みを行なっている間に取得されるバックアップ。「ウォームバックアップ」および「コールドバックアップ」と対比してください。

ホットバックアップでは、データファイルの単なるコピー以外にも多くの作業が行われます。バックアップの進行中に挿入または更新されたすべてのデータを含む必要があり、バックアップの進行中に削除されたすべてのデータを除外する必要があります。またトランザクションによって開始されたがコミットされていないすべての変更を無視する必要があります。

特に InnoDB テーブルに対して、また MyISAM およびほかのストレージエンジンからのテーブルに対しても、ホットバックアップを実行する Oracle 製品が MySQL Enterprise Backup です。

ホットバックアッププロセスは 2 つのステージから構成されます。InnoDB データファイルの最初のコピーで raw バックアップが生成されます。適用ステップで、バックアップの実行中に行われたデータベースへのすべての変更が組み込まれます。変更の適用により、準備されたバックアップが生成されます。これらのファイルは、必要な場合はいつでもリストアできる状態です。

完全バックアップは、InnoDB データをコピーするホットバックアップフェーズと、それに続く、MyISAM テーブルや .frm ファイルなどの非 InnoDB データをコピーするウォームバックアップフェーズから構成されます。

[適用](#)、[コールドバックアップ](#)、[.frm ファイル](#)、[完全バックアップ](#)、[InnoDB](#)、[インスタンス](#)、[MySQL Enterprise Backup](#)、[準備されたバックアップ](#)、[raw バックアップ](#)、[ウォームバックアップ](#)も参照

マ

マスター

レプリケーション構成で、更新をスレーブサーバーのセットに送信するデータベースサーバーです。一般に、そのリソースの大部分が書き込み操作専用となるため、ユーザークエリはスレーブに任せられます。MySQL Enterprise Backup では通常、マスターではなくスレーブサーバーでバックアップを実行し、システム全体の速度低下を最低限に抑えます。

[MySQL Enterprise Backup](#)、[レプリケーション](#)、[スレーブ](#)も参照

マニフェスト

バックアップに関する環境 (たとえばコマンド行引数) およびデータファイルの記録であり、それぞれ [meta/backup_create.xml](#) と [meta/backup_content.xml](#) のファイルに格納されています。このデータは、診断およびトラブルシューティング手順中に、管理ツールで使用できます。

メ

メディア管理ソフトウェア

テープバックアップのライブラリなど、バックアップメディアを管理するソフトウェアプログラムのクラス。Oracle Secure Backup がその一例です。MMS と略記されます。

[Oracle Secure Backup](#)も参照

ユ

行フォーマット

InnoDB テーブルからの行のディスクストレージフォーマット。InnoDB に圧縮などの新しい機能が用意されるのに伴い、ストレージの効率性とパフォーマンスの向上をサポートするために新しい行フォーマットが導入されます。

テーブルごとに独自の行フォーマットがあり、[ROW_FORMAT](#) オプションを通じて指定されます。各 InnoDB テーブルの行フォーマットを確認するには、コマンド [SHOW TABLE STATUS](#) を発行します。システムテーブルスペース内のすべてのテーブルは同じ行フォーマットを共有するため、ほかの行フォーマットを活用するには、通常、各テーブルが個別のテーブルスペースに格納されるように、[innodb_file_per_table](#) オプションを設定する必要があります。

リ

リストア

バックアップ操作の逆です。準備されたバックアップからのデータファイルが、データの問題を修復したり、システムを以前の状態に戻したりするために元の場所に戻されます。

[バックアップ](#)、[準備されたバックアップ](#)も参照

リポジトリ

サーバーリポジトリとバックアップリポジトリとを区別します。

[バックアップリポジトリ](#)、[サーバーリポジトリ](#)も参照

履歴テーブル

完了したバックアップ操作の詳細を保持するテーブル `mysql.backup_history`。バックアップジョブが実行している間、詳細 (特に変更ステータス値) は進行状況テーブルに記録されます。

[バックアップ](#), [進行状況テーブル](#)も参照

レ

レプリケーション

一連のスレーブサーバーと同期したマスターサーバーからのデータおよび DML 操作を伴う、MySQL 配備の一般的な構成。MySQL Enterprise Backup では、あるサーバー上でバックアップを取得し、別のシステム上でリストアすることで、データが用意された新しいスレーブサーバーを作成できます。マスターではなくスレーブサーバーからデータをバックアップして、システム全体の速度低下を最低限に抑えることもできます。

[マスター](#), [MySQL Enterprise Backup](#), [スレーブ](#)も参照

ロ

ログ

MySQL Enterprise Backup 製品内では、複数のタイプのログファイルが使用されます。もっとも一般的なものが、増分バックアップ中に参照される InnoDB Redo ログです。

[増分バックアップ](#), [Redo ログ](#)も参照

ログシーケンス番号

[LSN](#)も参照

ロック

[一時停止](#), [ウォームバックアップ](#)も参照

論理バックアップ

実際のデータファイルをコピーせずにテーブル構造とデータを再生成するバックアップ。たとえば、`mysqldump` コマンドは、データを再度作成できる `CREATE TABLE` や `INSERT` などのステートメントがその出力に含まれるため、論理バックアップを生成します。物理バックアップと対比してください。

[バックアップ](#), [物理バックアップ](#)も参照

索引

シンボル

- apply-log オプション, 58
- backup-dir オプション, 67
- cloud-access-key-id オプション, 93
- cloud-aws-region オプション, 93
- cloud-bucket オプション, 93
- cloud-object-key オプション, 93
- cloud-proxy オプション, 93
- cloud-secret-access-key オプション, 93
- cloud-service オプション, 93
- cloud-trace オプション, 93
- comments-file オプション, 70
- comments オプション, 70
- compress-level オプション, 32, 71
- compress-method オプション, 71
- compress オプション, 32, 70
- data_home_dir オプション, 66
- databases-list-file オプション, 78
- databases オプション, 78
- datadir オプション, 66
- decrypt オプション, 92
- disable-manifest オプション, 82
- dst-entry オプション, 81
- encrypt オプション, 92
- exclude-tables オプション, 75
- exec-when-locked オプション, 94
- include-tables オプション, 74
- include オプション, 33, 77
- incremental-backup-dir オプション, 74
- incremental-base オプション, 73
- incremental-with-redo-log-only オプション, 72
- incremental オプション, 72
- key-file オプション, 93
- key オプション, 93
- limit-memory オプション, 84
- log-bin-index, 86
- master-info-file, 87
- no-history-logging オプション, 70
- no-locking オプション, 84
- number-of-buffers オプション, 82
- on-disk-full オプション, 85
- only-innodb オプション, 76, 79
- only-known-file-types オプション, 75
- optimistic-busy-tables, 87
- optimistic-time, 87
- page-reread-count オプション, 85
- page-reread-time オプション, 85
- process-threads オプション, 83
- progress-interval, 92
- read-threads オプション, 83
- relay-log-index, 86
- relaylog-info-file, 87
- sbt-database-name オプション, 81
- sbt-environment オプション, 82
- sbt-lib-path オプション, 81
- show-progress, 89
- skip-binlog, 86
- skip-relaylog, 86
- skip-unused-pages, 85
- slave-info オプション, 94
- sleep オプション, 84
- src-entry オプション, 81
- start-lsn オプション, 74
- suspend-at-end オプション, 94
- uncompress オプション, 71
- use-tts オプション, 76
- with-timestamp オプション, 70
- write-threads オプション, 84
- .ARM ファイル, 139
- .ARZ ファイル, 139
- .CSM ファイル, 7
- .CSV ファイル, 7
- .frm ファイル, 7, 33, 139
- .ibd ファイル, 7, 47, 139
- .ibz ファイル, 7, 139
- .MRG ファイル, 139
- .MYD ファイル, 7
- .MYD ファイル, 139
- .MYI ファイル, 7
- .MYI ファイル, 139
- .opt ファイル, 7, 140
- .par ファイル, 7, 140
- .TRG ファイル, 7
- .TRG ファイル, 140
- .TRN ファイル, 7
- 圧縮, 142
- 圧縮バックアップ, 6, 32, 36, 44, 70, 129
- 圧縮レベル, 142
- 暗号化, 107
- 暗号化バックアップ, 107
- 一時停止, 143
- イメージ, 142
- インスタンス, 143
- ウォームバックアップ, 5, 143
- エラーコード, 111
- オプション、mysqlbackup, 52
 - mysql で一般的なオプション, 63
 - 圧縮の, 70
 - 暗号化の, 92
 - オーバーヘッド、パフォーマンス、スケーラビリティの制御の, 82
 - クラウドストレージの, 93
 - 構成ファイル内, 95
 - 進行状況レポートの制御の, 89
 - 接続, 64
 - 操作のモード, 57
 - 増分バックアップの, 71
 - 単一ファイルバックアップの, 80
 - データベースファイルのレイアウト, 65
 - 特別なバックアップの種類, 94
 - バックアップファイルのレイアウト, 67
 - 標準オプション, 63
 - 部分バックアップの, 74
 - メタデータの生成用, 70
 - メッセージロギングの制御の, 88
- オフライン, 143

オンライン, 143
完全バックアップ, 29, 125, 143
クラウドバックアップ, 39
クラッシュリカバリ, 43, 143
権限, 25
構成オプション, 95
構成ファイル, 144
コールドバックアップ, 5, 143
コマンド行ツール, 6
サーバー, 144
サーバーリポジトリ, 144
システムテーブルスペース, 7, 144
準備されたバックアップ, 7, 43, 144
除外, 144
進行状況インジケータ, 89
進行状況テーブル, 144
ストリーミング, 39, 144
ストリーミングバックアップ, 6
スレーブ, 97, 98, 144
正規表現, 145
接続, 145
接続オプション, 64
増分バックアップ, 6, 71, 129, 145
単一ファイルバックアップ, 6, 62, 80, 145
抽出, 145
抽出オプション, 80
停止時間, 146
データディクショナリ, 146
データベース, 146
テープバックアップ, 39, 109
テーブル, 145
テーブルスペース, 146
適用, 146
特定の時点, 146
バイナリログ, 46, 146
破損の問題, 111
バックアップ, 147
 InnoDB テーブルのみ, 51
 raw, 7, 43
 圧縮, 6, 32, 36, 44, 70, 129
 暗号化, 107
 ウォーム, 5
 完全, 29, 125
 クラウドへの, 39
 検証, 28
 コールド, 5
 準備された, 7, 43
 進行状況レポート, 89
 スケジュール, 42
 ストリーミング, 6, 39
 増分, 6, 30, 71, 129
 単一ファイル, 6
 テープへの, 39, 109
 トラブルシューティング, 111
 非圧縮, 6, 36
 物理, 6
 部分, 33, 74
 並列, 6
 ホット, 5
 メッセージロギング, 88
 モニタリング, 111
 リストアの準備, 43
 論理, 6
バックアップされるファイル, 7
バックアップジョブのモニタリング, 111
バックアップ操作のパフォーマンス, 6
バックアップデータのディスクストレージ, 6, 39
バックアップデータの領域, 6
バックアップの検証, 28, 60
バックアップのトラブルシューティング, 111
バックアップのリストア, 43
 -use-fts オプションで作成されたバックアップ, 46
 mysqlbackup オプション, 59
 概要, 15
 準備, 43
 単一の .ibd ファイル, 47
 ポイントインタイム・リカバリ, 46
 元の場所, 28
 例, 44
バックアップリポジトリ, 147
パフォーマンス
 バックアップの, 101
 リストアの, 104
非圧縮バックアップ, 6, 36
ファイル形式, 147
復号化, 107
含める, 147
物理的バックアップ, 6
物理バックアップ, 147
部分バックアップ, 33, 74, 147
部分リストア, 147
並列バックアップ, 6, 101, 104, 147
変更
 リリースノート, 131
ベンチマーク, 101
ポイントインタイム・リカバリ, 46
ホットバックアップ, 5, 147
マスター, 98, 148
マニフェスト, 7, 82, 114, 148
メタディレクトリ, 7
メッセージロギング, 88
メディア管理ソフトウェア, 148
メディア管理ソフトウェア (MMS) 製品, 109
行フォーマット, 148
よくある質問, 115
リストア, 148
リポジトリ, 148
リリースノート, 131
履歴テーブル, 149
レプリケーション, 97, 98, 98, 149
ログ, 7, 58, 149
 バックアップ操作の, 112
ロック, 149
論理バックアップ, 6, 149

A

Antelope, 140
Antelope, 51
apply-incremental-backup オプション, 44, 58

B

backup-and-apply-log オプション, 58
backup-dir-to-image オプション, 62
backup-image オプション, 80
backup-my.cnf, 140
backup-my.cnf ファイル, 7
backup-to-image オプション, 58, 80
backup_content.xml, 7
backup_content.xml ファイル, 114
backup_create.xml, 7
backup_create.xml ファイル, 114
BACKUP_HISTORY テーブル, 112
BACKUP_PROGRESS テーブル, 112
backup_variables.txt ファイル, 7
backups
 オーバーヘッド、パフォーマンス、スケーラビリティの制御, 82
backup オプション, 58
Barracuda, 140
Barracuda, 51
binlog, 140

C

ChangeLog, 131
comments.txt ファイル, 7, 70
copy-back-and-apply-log オプション, 59
copy-back オプション, 15, 28, 43, 59
cron ジョブ, 42

D

datadir ディレクトリ, 7

E

extract オプション, 63

F

FAQ, 115

G

GRANT ステートメント, 25

I

ib_logfile ファイル, 7
ibbackup_logfile ファイル, 7
ibdata ファイル, 7, 140
ibreset コマンド, 112
image-to-backup-dir オプション, 62, 80, 80
image_files.xml ファイル, 7, 114
InnoDB, 141
InnoDB テーブル, 5, 7, 51, 51
 圧縮バックアップ機能, 32
 増分バックアップ機能, 30

L

list-image オプション, 63, 80
LSN, 141

LSN, 30, 71

M

MEMORY テーブル, 41
MMS 製品, 109
my.cnf, 141
my.ini, 141
MyISAM, 141
MyISAM テーブル, 51
MySQL Enterprise Backup, 141
MySQL Enterprise Backup のインストール, 17
MySQL Enterprise Monitor, 111
mysqlbackup, 141
mysqlbackup, 51
 オプション, 52
 およびメディア管理ソフトウェア (MMS) 製品, 109
 概要, 6
 構成オプション, 95
 使用, 23
 生成されるファイル, 7
 操作のモード, 57
 必要な権限, 25
 例, 29
mysqlbinlog コマンド, 46
mysqldump, 141
mysqldump, 41

O

Oracle Secure Backup, 141
OSB, 142

P

posix_fadvise() システムコール, 6

R

RAID, 101, 104
raw バックアップ, 7, 43, 142
Redo ログ, 142

S

SBT, 142
System Backup to Tape, 142

V

validate オプション, 61

