



# MySQL入門(バックアップ編)

Yoshiaki Yamasaki / 山崎 由章

MySQL Senior Sales Consultant, Asia Pacific and Japan

# SAFE HARBOR STATEMENT

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。  
また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。  
以下の事項は、マテリアルやコード、機能を提供することをコミットメントするものではない為、  
購買決定を行う際の判断材料になさらないで下さい。

オラクル製品に関して記載されている機能の開発、リリースおよび時期については、  
弊社の裁量により決定されます。

# Program Agenda

- 1 ▶ バックアップ手法の概要
- 2 ▶ バックアップ運用設計時の主な考慮事項
- 3 ▶ バックアップ対象のファイル
- 4 ▶ バックアップ/リストア例
- 5 ▶ MySQL Enterprise Backup

# Program Agenda

- 1 ▶ バックアップ手法の概要
- 2 ▶ バックアップ運用設計時の主な考慮事項
- 3 ▶ 対象のファイル
- 4 ▶ バックアップ/リストア例
- 5 ▶ MySQL Enterprise Backup

# バックアップ手法の違い

- ホットバックアップ or コールドバックアップ
- 物理バックアップ or 論理バックアップ
- フルバックアップ or 差分/増分バックアップ
  
- ※スタンバイサイトを利用する
  - ホットスワップとしての利用
  - バックアップ取得先としての利用

# ホットバックアップ or コールドバックアップ

- ホットバックアップ(オンラインバックアップ)
  - データベースを停止せず、稼働したままでバックアップデータを取得する
- コールドバックアップ(オフラインバックアップ)
  - データベース停止中にバックアップデータを取得する

# ホットバックアップ or コールドバックアップ

- ホットバックアップ(オンラインバックアップ)の例
  - トランザクションの仕組みを利用してバックアップを取得
    - mysqldumpでInnoDBテーブルをバックアップする
  - ロックを利用してバックアップを取得
    - mysqldumpでMyISAMテーブルをバックアップする
  - OSやハードウェアのスナップショットを利用する
    - LVMでスナップショットを取得してバックアップする(※)
  - 独自の方法でバックアップを取得する
    - MySQL Enterprise Backup(旧名称: InnoDB Hot Backup)でバックアップする(※)

※参考: 漢(オトコ)のコンピュータ道 より

MySQLバックアップ頂上決戦!! LVMスナップショット vs InnoDB Hot Backup

<http://nippondanji.blogspot.jp/2009/12/mysql-lvm-vs-innodb-hot-backup.html>

# ホットバックアップ or コールドバックアップ

- コールドバックアップ(オフラインバックアップ)の例
  - MySQLサーバをシャットダウンして、データディレクトリ以下のディレクトリとファイルを全てOSのコマンドでコピー



# 物理バックアップ or 論理バックアップ

- 物理バックアップ

- 物理的なファイルのバックアップ
- OSファイルのコピー、MySQL Enterprise Backup等で取得可能
- 利点: 最小限のサイズで取得できる  
バックアップ/リストアの速度が速い
- 欠点: バックアップ/リストアの単位はツールしだい  
異機種間、バージョン間で互換性が取れない場合がある(※)

## ※主な注意事項

- テーブル名に日本語文字を使用している場合
- 浮動小数点が混じっている場合
- テーブル名に大文字/小文字が混在している場合

# 物理バックアップ or 論理バックアップ

- 論理バックアップ

- データベースからデータを抜き出してバックアップする
- MySQLの場合は、SQLベースのバックアップを取得可能
- mysqldumpで取得可能
- 利点 : バックアップファイルを編集できる  
移植性が高い(他バージョン、他のRDBMS)
- 欠点 : 物理バックアップに比べてサイズが大きくなる  
バックアップ、リストアに時間がかかる(バイナリ⇔テキストの変換が入るため)

# フルバックアップ or 差分/増分バックアップ

- フルバックアップ(全体バックアップ)
  - データベース全体をバックアップする
- 差分/増分バックアップ(部分バックアップ)
  - 直近のバックアップ以降に更新されたデータのみをバックアップする
  - 差分バックアップ
    - 直近のフルバックアップ以降に更新されたデータをバックアップ
  - 増分バックアップ
    - 直近のバックアップ(種別はフルとは限らない)以降に更新されたデータをバックアップする

# フルバックアップ or 差分/増分バックアップ

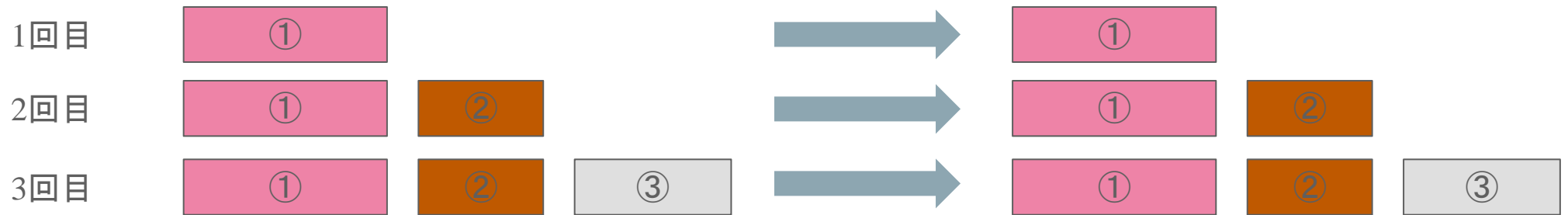
- フルバックアップ(全体バックアップ)

- 利点

- リストア処理が単純  
(バックアップデータが1カ所にまとまっているため)

- 欠点

- バックアップにかかる時間が長くなる (データベース全体をバックアップするため)
    - バックアップデータのサイズが大きくなる  
(データベースに対する更新量が少なくても、データベース全体をバックアップする)



# フルバックアップ or 差分/増分バックアップ

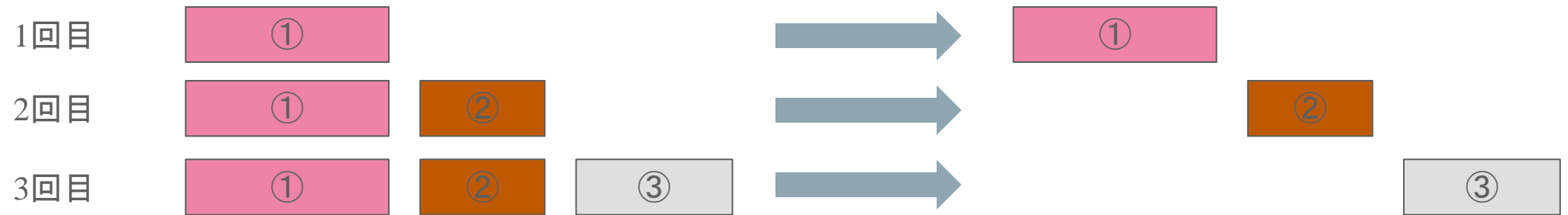
- 差分/増分バックアップ(部分バックアップ)

- 利点

- バックアップ時間が短くなる (更新したデータだけを対象にするため)
- バックアップデータのサイズが小さくなる (更新したデータだけを対象にするため)

- 欠点

- リストア処理の手順が複雑になる (フルバックアップと部分バックアップを使ってリストア)



# フルバックアップ or 差分/増分バックアップ

- MySQLでの差分/増分バックアップ手法
  - バイナリログを増分バックアップとして利用する
  - MySQL Enterprise Backup(有償ツール)で  
InnoDBデータの増分/差分バックアップを取得する

# ポイントインタイムリカバリとバイナリログ

## • ポイントインタイムリカバリ

- 特定の日時の状態にデータを復旧すること (例: 障害発生直前の状態まで復旧する)
- MySQLでは、バックアップファイルとバイナリログファイルを利用して、ポイントインタイムリカバリが可能 (バックアップファイルに対して、バイナリログファイルを使ってロールフォワードリカバリする)

## • バイナリログ

- 発行されたクエリのうち、更新系のSQL文のみを記録しているログファイル
- “--log-bin” オプションを設定することで出力できる(デフォルトでは出力されていない)
- コミット時にバイナリログに同期書き込みするためには、“sync\_binlog=1” を設定する

※バックアップを取得しているだけでは、バックアップ取得時点のデータしか復旧できないが、バイナリログを出力してリカバリに利用することで、特定時点のデータを復旧可能

# スタンバイサイトを利用する

- レプリケーション機能を利用して、スタンバイサイトを構築しておき、障害発生時はフェイルオーバーする

## – 利点

- 復旧時間が短い

## – 欠点

- 復旧ポイントは障害発生直前の状態のみ(※)
- 人的ミスに対する対策にはならない(※)

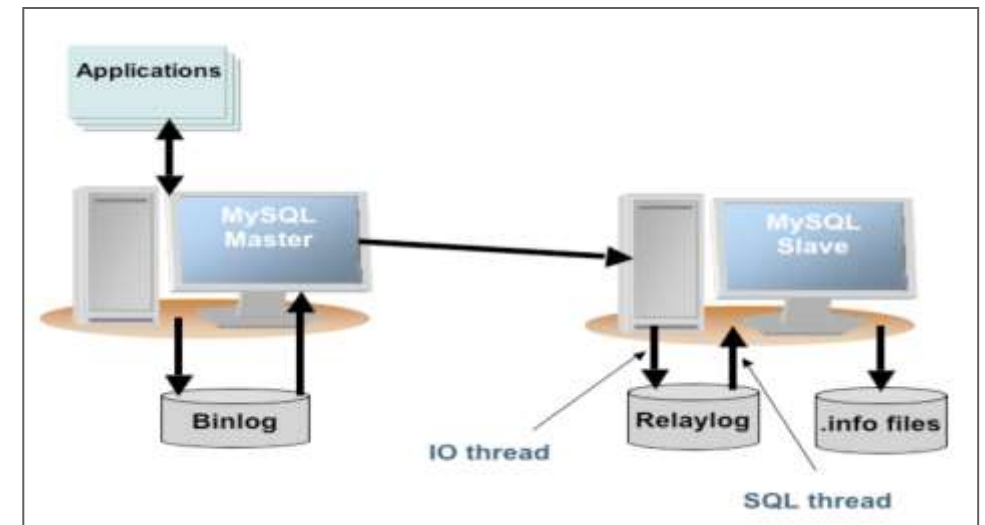
※MySQL5.6では、遅延レプリケーション機能で、意図的にスタンバイサイトを遅延させることも可能

- スタンバイサイトをバックアップ取得先として利用することも可能  
(本番環境に影響を与えずにバックアップ取得可能)



# レプリケーション

- MySQLの標準機能
  - シンプルな設定で利用可能
  - 多数のWebサイトで実績あり
- 同期方式：非同期 or 準同期
- 特長
  - 参照性能を向上させる構成
  - バックアップ用途でも利用可能
  - バイナリログを利用して、更新内容をスレーブに伝搬



# Program Agenda

- 1 ➤ バックアップ手法の概要
- 2 ➤ バックアップ運用設計時の主な考慮事項
- 3 ➤ 対象のファイル
- 4 ➤ バックアップ/リストア例
- 5 ➤ MySQL Enterprise Backup

# バックアップ運用設計時の主な考慮事項

- RTO (Recovery Time Objective) : 目標復旧時間
  - 障害発生時に「いつまでに復旧するか」という目標時間
  - 例 : データベースを1時間以内に復旧する必要がある
    - ⇒バックアップのリストア(再配置)+リカバリ(復旧)が1時間以内に終わる手法を採用する必要がある
- RPO (Recovery Point Objective) : 目標復旧時点
  - 障害発生時に「どの時点までのデータを復旧する必要があるのか」という目標時点
  - 例 : 前日のバックアップ取得時点まででいい
    - ⇒バックアップのリストアだけできれば要件を満たせるので、バイナリログの保全是必須では無い

# バックアップ運用設計時の主な考慮事項

- バックアップの保存期間、保存場所
  - バックアップを何世代保管するか？ (最低2世代は必要)
  - どこに保管するか？ (古いバックアップはテープ等に移して保管する、など)  
⇒DBサイズが大きいと、ストレージの必要容量に大きな影響を与える
- バックアップ取得頻度、手法
  - システム要件(例:システム稼働時間)やRTO、RPO等を考慮して、決定する

# Program Agenda

- 1 ➤ バックアップ手法の概要
- 2 ➤ バックアップ運用設計時の主な考慮事項
- 3 ➤ 対象のファイル**
- 4 ➤ バックアップ/リストア例
- 5 ➤ MySQL Enterprise Backup

# バックアップ対象

- データディレクトリ または データ全体
  - フルバックアップ
  - 論理 または 物理バックアップ
- ログファイル(バイナリログ)
  - 増分バックアップ
  - ポイントインタイムリカバリに利用
- 設定ファイル
  - `my.cnf`



# データディレクトリ

- データベース内容、ログ、およびステータスファイルの格納先
- デフォルトのディレクトリはインストール形式による
  - `/usr/local/mysql/data/` (tar形式)
  - `/var/lib/mysql` (RPMパッケージ)
- サーバ起動オプションで設定可能
  - `datadir=/path/to/datadir/`
- サーバが利用しているディレクトリは下記コマンドで確認可能
  - `mysql> SHOW VARIABLES like 'datadir';`

※InnoDB関連ファイルのパスをデータディレクトリ以外に設定している場合は、バックアップ時にそれらも取得する  
(デフォルト設定では、InnoDB関連ファイルはdatadir配下に出力されている)

## 補足: InnoDB関連のパス

- データファイル(ibdata)のパス
  - innodb\_data\_home\_dir + innodb\_data\_file\_path
- REDOログファイルのパス
  - innodb\_log\_group\_home\_dir
- Undoログ用のテーブルスペースのパス
  - innodb\_undo\_directory (※)

※指定したディレクトリにUndoログを出力するためには、innodb\_undo\_tablespacesも合わせて設定が必要



# バイナリログ

- 発行されたクエリのうち、更新系のSQL文のみを記録しているログファイル
  - クエリ実行日時などのメタデータも記録
  - トランザクションのコミット時に同期的に記録 (`sync_binlog=1`)
- バイナリ形式で記録
  - `mysqlbinlog` コマンドにてテキスト化が可能
- 起動オプションを指定して、出力する
  - `---log-bin[=file_name]`
  - 通常の運用時には利用することを推奨
  - データディレクトリとは別のディスクに出力することを推奨
- ログファイル名の拡張子に通し番号を記録
  - 例) `file_name-bin.001`, `file_name-bin.002`, etc.
  - 現在利用中のログ番号はインデックスファイルに記録 (`file_name.index`)

# バイナリログの管理

- **SHOW MASTER STATUS** コマンドで現在使用中のバイナリログファイル名とポジションを確認
- **SHOW MASTER LOGS** コマンドで全てのバイナリログファイル名を列挙
- **FLUSH [BINARY] LOGS** コマンドまたはMySQLサーバの再起動でログファイルのローテーション
- **PURGE MASTER** コマンドで特定の時点までのバイナリログを削除
- **RESET MASTER** コマンドで全てのバイナリログを削除

# バイナリログの管理

```
mysql> SHOW MASTER STATUS;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
MySQL.000007	107		

1 row in set (0.00 sec)

```
mysql>
```

```
mysql> SHOW MASTER LOGS;
```

Log_name	File_size
MySQL.000001	1110
MySQL.000002	2797
...	
MySQL.000007	107

7 rows in set (0.00 sec)

# バイナリログの管理

```
mysql> FLUSH BINARY LOGS;  
Query OK, 0 rows affected (0.42 sec)
```

```
mysql> SHOW MASTER STATUS;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
MySQL.000008	107		

1 row in set (0.00 sec)

```
mysql> SHOW MASTER LOGS;
```

Log_name	File_size
MySQL.000001	1110
MySQL.000007	146
MySQL.000008	107

8 rows in set (0.00 sec)

# バイナリログの管理

```
mysql> PURGE MASTER LOGS TO 'MySQL.000003';  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SHOW MASTER LOGS;
```

Log_name	File_size
MySQL.000003	2315
MySQL.000004	628
MySQL.000005	1090
MySQL.000006	126
MySQL.000007	146
MySQL.000008	107

```
6 rows in set (0.00 sec)
```

# バイナリログの管理

```
mysql> RESET MASTER;  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SHOW MASTER LOGS;
```

```
+-----+-----+  
| Log_name      | File_size |  
+-----+-----+  
| MySQL.000001 |      107 |  
+-----+-----+  
1 row in set (0.00 sec)
```

## 補足：バイナリログのリモートバックアップ (MySQL 5.6~)

- `mysqlbinlog` コマンドを使って、リモートサーバのバイナリログをコピー可能
  - 最新のバイナリログをコピーし続けることも可能
  - 接続先ホストを自ノードに設定すれば、自ノード内でバイナリログをコピーすることも可能

# 補足: バイナリログのリモートバックアップ (MySQL 5.6~)

## • 実行例

- 指定したバイナリログ(binlog.000130~000132)をコピー

```
mysqlbinlog --read-from-remote-server --host=host_name  
--raw binlog.000130 binlog.000131 binlog.000132
```

- 指定したファイル以降に生成されたバイナリログを全てコピー

```
mysqlbinlog --read-from-remote-server --host=host_name  
--raw --to-last-log binlog.000130
```

- 指定したファイル以降に生成されたバイナリログを全てコピーし、その後生成されたファイルもコピーし続ける

```
mysqlbinlog --read-from-remote-server --host=host_name  
--raw --stop-never binlog.000130
```



# 補足: ストレージエンジンごとの特性について

- InnoDB

- 共有テーブルスペースファイルとInnoDBログファイルの組で1つの単位。  
`innodb_file_per_table`設定時は\*.ibd も必要
  - 共有テーブルスペースファイル: `ibdata1`、`ibdata2`、...
  - InnoDBデータファイル: `.ibd`ファイル
  - InnoDBログファイル: `ib_logfile0`、`ib_logfile1`、...
  - テーブル定義ファイル: `.frm`ファイル
- MySQL5.6ではトランスポータブル表領域機能を利用して、テーブル単位でファイルコピーして移動可能
  - MySQL 5.5以前のバージョンではファイルコピーだけでは移動は出来ない
- クラッシュセーフ、トランザクション対応
- ロックは行単位
- MySQL Enterprise Backup(旧名称 InnoDB Hot Backup)でホットバックアップ可能

# 補足: ストレージエンジンごとの特性について

- MyISAM
  - 3つのファイルから構成される(.frm、.MYD、.MYI)
  - ファイルとして書込み禁止(FLUSH TABLES WITH READ LOCK)すればファイルコピーできる
  - クラッシュセーフでは無い  
(MySQLサーバやOSのクラッシュ時にはrepairが必要)
  - ロックはテーブル単位

# Program Agenda

- 1 ➤ バックアップ手法の概要
- 2 ➤ バックアップ運用設計時の主な考慮事項
- 3 ➤ 対象のファイル
- 4 ➤ バックアップ/リストア例
- 5 ➤ MySQL Enterprise Backup

# 前提

- 設定ファイル `my.cnf` の配置先 `/usr/local/mysql/data/my.cnf`

```
[mysqld]
datadir=/usr/local/mysql/data
socket=/usr/local/mysql/data/mysql.sock
user=mysql
log-bin=MySQL
```

# OSコマンドによる物理バックアップ例

## 1. MySQLサーバを停止

```
-$ mysqladmin --user=root --password=root ¥  
--socket=/usr/local/mysql/data/mysql.sock shutdown
```

## 2. コールドバックアップを取得する

```
-$ cp -rp /usr/local/mysql/data /backup/mysql-20150623
```

## 3. MySQLサーバを起動する

```
-$ mysqld --defaults-file=/usr/local/mysql/data/my.cnf &
```

# 物理バックアップのリストア例

1. マシンやディスクが壊れている場合は復旧する
2. MySQLサーバが起動している場合は停止する
3. 既存のデータベース領域を削除する
  - \$ `rm -rf /usr/local/mysql/data`
4. バックアップファイルをリストアする
  - \$ `cp -rp /backup/mysql-20150623 /usr/local/mysql/data`
5. MySQLサーバを起動する
  - \$ `mysqld --defaults-file=/usr/local/mysql/data/my.cnf &`

# mysqldumpによるバックアップ例

- 全てのテーブルをロックしてデータベース全体のバックアップを取得

```
– $ mysqldump --user=root --password=root --master-data=2 ¥  
    --socket=/usr/local/mysql/data/mysql.sock ¥  
    --hex-blob --default-character-set=utf8 --all-databases ¥  
    --lock-all-tables > mysql_bkup_dump.sql
```

- InnoDBのトランザクションを使用してデータベース全体のバックアップを取得

```
– $ mysqldump --user=root --password=root --master-data=2 ¥  
    --socket=/usr/local/mysql/data/mysql.sock ¥  
    --hex-blob --default-character-set=utf8 --all-databases ¥  
    --single-transaction > mysql_bkup_dump.sql
```

# mysqldumpのオプション

- **--master-data=2**
  - バックアップ取得のバイナリファイル名とバイナリファイル内の位置(Position)をコメントとしてバックアップファイルに記録
- **--hex-lob**
  - バイナリ型(BINARY、VARBINARY、BLOB)とBIT型のデータを16進数表記で出力
- **--default-character-set**
  - mysqldumpがデフォルトで利用するキャラクタセットを指定。  
通常はMySQLサーバのシステム変数default-character-setと同じものを指定すれば良い
- **--all-databases**
  - 全てのデータベースをバックアップ
- **--lock-all-tables**
  - 全てのテーブルをロックしてバックアップを取得する
- **--single-transaction**
  - InnoDBがサポートしているトランザクションの仕組みを利用して、InnoDBテーブルに限り一貫性のとれたバックアップを取得する



# 注意事項: mysqldumpによるバックアップ

- データの整合性を保つために、バックアップ取得中は、テーブルに関するDDL文(※)を実行しないこと

※ALTER TABLE, CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE

- マニュアルの“--single-transaction”オプションの説明部分より引用
  - 「While a --single-transaction dump is in process, to ensure a valid dump file (correct table contents and binary log coordinates), no other connection should use the following statements: ALTER TABLE, CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE.」

# mysqldumpからのリストア例

1. マシンやディスクが壊れている場合は復旧する
2. MySQLサーバが起動している場合は停止する
3. 既存のデータベース領域を削除後、再作成する  
(必要に応じて事前にバックアップを取得)
  - \$ `rm -rf /usr/local/mysql/data`
  - \$ `mkdir /usr/local/mysql/data`
4. バイナリログの出力を停止 (`my.cnf`から`log-bin`をコメントアウト)
  - ※この例の場合は、バックアップしておいた `my.cnf` を  
`/usr/local/mysql/data` 配下に配置してから、`log-bin`をコメントアウト

# mysqldumpからのリストア例

5. **権限テーブル**と**ネットワーク接続**を無効化した状態でMySQLサーバを起動

```
– $ mysqld --defaults-file=/usr/local/mysql/data/my.cnf ¥  
  --skip-networking --skip-grant-tables &
```

6. バックアップファイルに記述されたSQL文を実行

```
– $ mysql --default-character-set=utf8 ¥  
  --socket=/usr/local/mysql/data/mysql.sock < mysql_bkup_dump.sql
```

7. バイナリログの出力を再開して、正常に再起動

```
– $ mysqladmin --user=root --password=root ¥  
  --socket=/usr/local/mysql/data/mysql.sock shutdown  
– $ mysqld --defaults-file=/usr/local/mysql/data/my.cnf &
```

# mysqldumpからのリストアの注意点

- ※MySQL 5.5以上の場合、この手順では性能統計情報を格納する `performance_schema` が作成されず、エラーログに `performance_schema` が無いというエラーが出力される。

- [回避策]

1. サーバ再起動前に、`performance_schema`ディレクトリを `datadir` 配下にコピーする  
(インストールディレクトリ/`data/performance_schema`)
2. MySQLサーバ再起動後、`mysql_upgrade`を実行する
  - \$ `mysql_upgrade --user=root --password=root ¥`  
`--socket=/usr/local/mysql/data/mysql.sock`

# バイナリログを使用したポイントインタイムリカバリ

## 1. ネットワーク接続を無効化した状態でMySQLサーバを起動

```
– $ mysqld_safe --defaults-file=/usr/local/mysql/data/my.cnf ¥  
  --skip-networking &
```

## 2. バイナリログからロールフォワード用SQL文を生成し、生成したSQL文を実行する(ファイル名とポジションは、適切なものを指定)

```
– $ mysqlbinlog --disable-log-bin --start-position=1017 ¥  
  MySQL.000010 MySQL.000011 > recover.sql
```

```
– $ mysql --user=root --password=root ¥  
  --socket=/usr/local/mysql/data/mysql.sock ¥  
  --default-character-set=utf8 < recover.sql
```

# バイナリログを使用したポイントインタイムリカバリ

## 3. 正常に再起動

- \$ `mysqladmin --user=root --password=root ¥`  
`--socket=/usr/local/mysql/data/mysql.sock shutdown`
- \$ `mysqld --defaults-file=/usr/local/mysql/data/my.cnf &`

# Program Agenda

- 1 バックアップ手法の概要
- 2 バックアップ運用設計時の主な考慮事項
- 3 対象のファイル
- 4 バックアップ/リストア例
- 5 MySQL Enterprise Backup

# MySQL Enterprise Backup

## 高速、オンラインバックアップ & リカバリ

- MySQL Enterprise Editionで使用可能なInnoDBのオンラインバックアップツール
- フル、増分、部分バックアップ(圧縮可能)
- ポイントインタイム、フル、部分リカバリ
- マルチスレッドによる並列バックアップ&リカバリ処理
- クラウドストレージとの直接の連携(S3, etc.)
- 暗号化 – AES 256
- バイナリログおよびリレーログのバックアップ
- Oracle Secure Backupとの連携
- マルチプラットフォーム対応 (Windows, Linux, Unix)

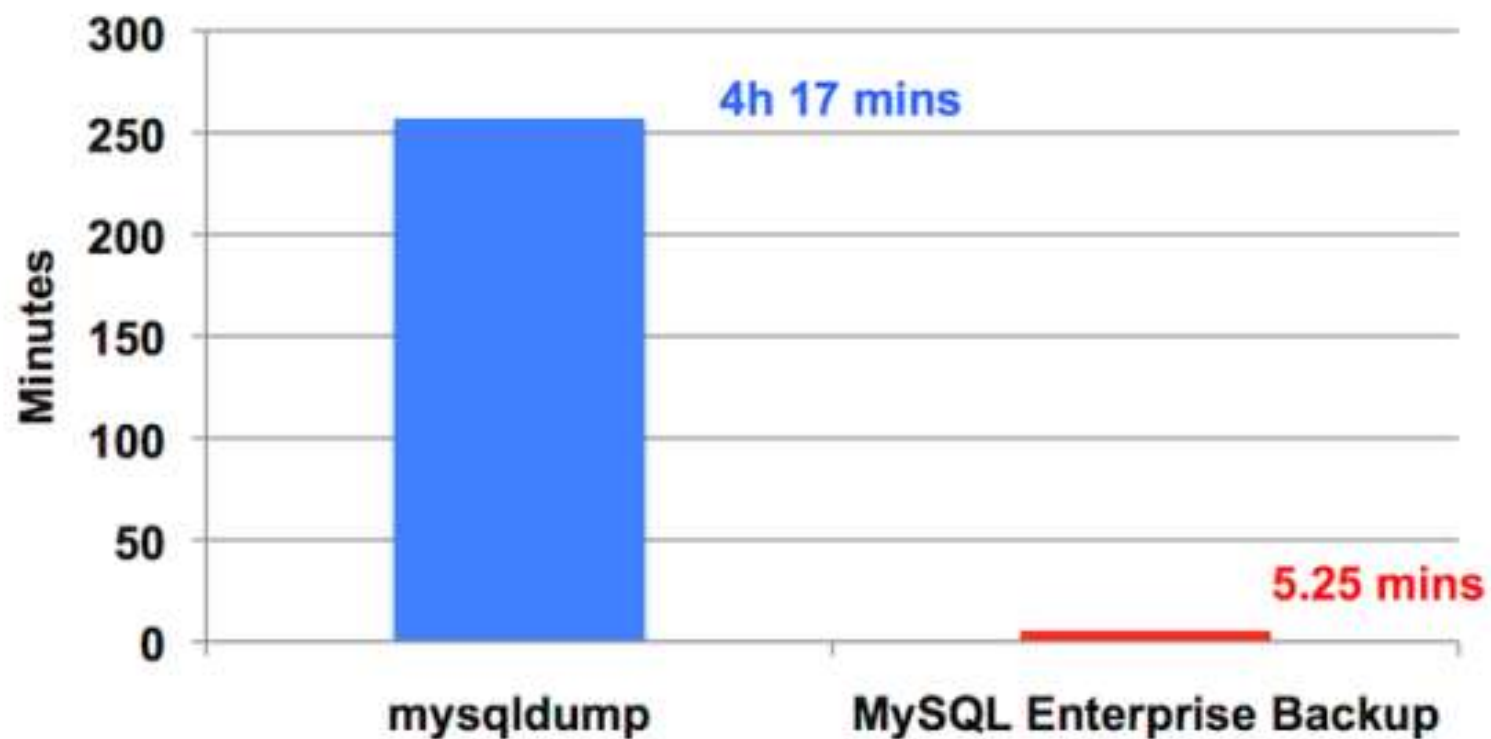


## MySQL Enterprise Backup の特徴と利点



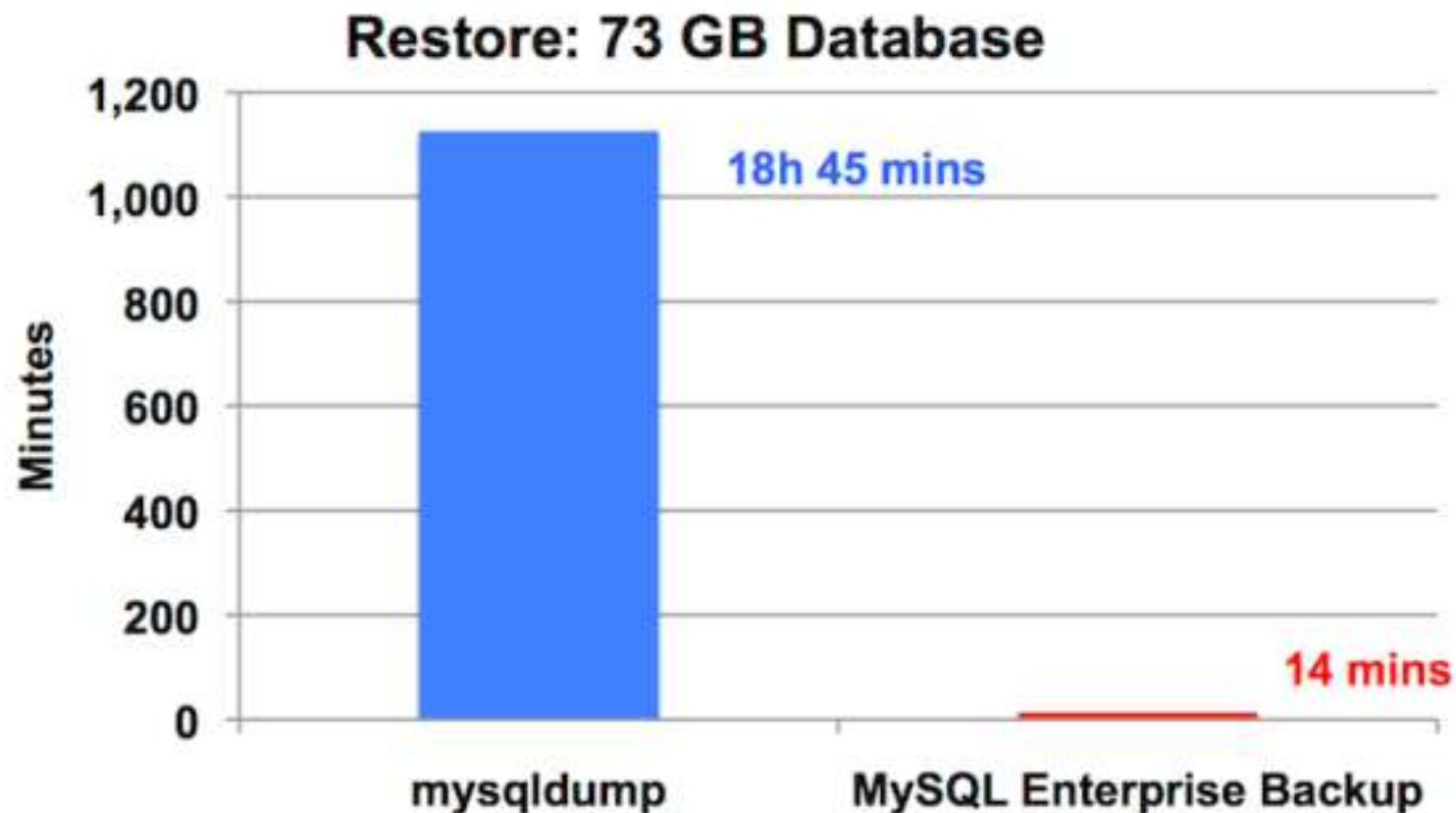
# 高速なバックアップ

Backup: 73 GB Database



mysqldumpより49倍速い

# 高速なリストア



**mysqldumpより80倍速い  
⇒ RTOが短い場合に役立つ**

# MySQL Enterprise Backupの動作

- ステップ1: InnoDBのデータファイル(ibdata&.ibd)をバックアップ
  - この時点のバックアップは、DB全体として整合性が取れていない  
(バックアップ取得中にもデータが更新されている可能性があるため)
- ステップ2: InnoDBのログファイル(ib\_logfile)をバックアップ
  - データファイルバックアップ中のログファイルをバックアップ
  - ログファイルにはInnoDBへの更新内容が記録されている
- ステップ3: バックアップしたログファイルをデータファイルに適用
  - DB全体として整合性が取れた状態が完成
- ステップ4: リストア(再配置)
  - 整合性のとれたバックアップをリストア

# バックアップオプション

- 圧縮バックアップ: `--compress`, `--compress-level=LEVEL`
- SSH経由のストリーム・バックアップ
- テープデバイスへの直接バックアップ、SBTインターフェース利用
- 部分バックアップ: `--include`, `--with-tts`
- 増分バックアップ
- 未使用領域のスキップ: `--skip-unused-pages`
- クラウドストレージとの直接連携(S3, etc)
- 暗号化(AES256 )

# バックアップ操作(基本)

- バックアップ取得とログ適用を1回の操作で実行

```
shell# mysqlbackup --user=root -p --backup-dir=/home/admin/backups ¥  
> backup-and-apply-log
```

- バックアップ取得とログ適用を分けて実行

```
shell# mysqlbackup --user=root -p --backup-dir=/home/admin/backups ¥  
> backup
```

```
shell# mysqlbackup --user=root -p --backup-dir=/home/admin/backups ¥  
> apply-log
```

# リストア操作(基本)

- ログ適用済みのバックアップをリストア

```
shell# mysqlbackup --defaults-file=/usr/local/mysql/my.cnf ¥  
> --backup-dir=/home/admin/backups ¥  
> copy-back
```

- ログ適用前のバックアップにログを適用し、リストア

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

# バックアップオプション: 圧縮

- データによっては80%以上の大幅な圧縮も可能

```
shell# mysqlbackup --backup-dir=/opt/mysql/backups/uncompressed ¥  
> backup 2>/dev/null && du -csh /opt/mysql/backups/uncompressed  
334M    /opt/mysql/backups/uncompressed  
334M    total
```

```
shell# mysqlbackup --backup-dir=/opt/mysql/backups/compressed ¥  
> --compress --compress-level=9 backup 2>/dev/null && du -csh ¥  
> /opt/mysql/backups/compressed  
81M    /opt/mysql/backups/compressed  
81M    total
```

# バックアップオプション: ストリーム・バックアップ

- リモートサーバにバックアップデータを出力
- テンポラリディレクトリを一時出力先として指定

```
shell# mysqlbackup --backup-image=- --backup-dir=/tmp/backup ¥  
> --disable-manifest backup-to-image 2>/dev/null | ssh matt@solo ¥  
> "cat > /tmp/backup.img"
```

```
shell# ssh matt@solo "ls -lh /tmp/backup.img"  
-rw-r--r--  1 matt      staff      333M Sep 10 15:54 /tmp/backup.img
```



# バックアップオプション: SBTインターフェースの利用

- 既存のシステムバックアップツールとの統合可能
  - Oracle Secure Backup (fully supported)
  - IBM Tivoli Storage Manager
  - Symantec NetBackup

```
shell-osb# mysqlbackup --backup-image=sbt:backup-mattprod-2013-09-08 ¥  
> --backup-dir=/tmp/backup backup-to-image
```

```
shell-tsm# mysqlbackup --backup-image=sbt:my-tsm-backup ¥  
> --sbt-lib-path=/usr/lib/libobk.so ¥  
> --sbt-environment="TDPO_OPTFILE=/opt/ibm/tsm/tdpo.opt" ¥  
> --backup-dir=/tmp/backup backup-to-image
```

# バックアップオプション: 部分バックアップ

- 重要なテーブル/スキーマのみをバックアップ
- InnoDBのトランスポートابل・テーブルスペース(tss)もサポート

```
shell# mysqlbackup --include=sakila.* --only-innodb-with-frm=related ¥  
> backup
```

```
shell# mysqlbackup --use-tts=with-minimum-locking ¥  
> --include=employees.* --backup-dir=/opt/mysql/backups backup
```

# バックアップオプション: 増分バックアップ

- 前回のフルバックアップからの変更点のみをバックアップ

```
shell# mysqlbackup --incremental ¥  
> --incremental-base=dir:/opt/mysql/backup/monday ¥  
> --incremental-backup-dir=/opt/mysql/backup/tuesday backup
```

```
shell# mysqlbackup --incremental ¥  
> --incremental-base=history:last_backup --with-timestamp ¥  
> --incremental-backup-dir=/opt/mysql/backup backup
```

# バックアップオプション: 未使用領域のスキップ

- InnoDBの空白領域や未使用領域はバックアップしない
- このオプションと圧縮を組み合わせることでバックアップの効率化

```
shell# mysqlbackup --backup-dir=/opt/mysql/backups/compressed ¥  
> --compress --compress-level=9 --skip-unused-pages backup
```

# バックアップオプション: クラウドストレージとの直接連携

- Amazon S3上へバックアップを取得  
(MEB3.11時点に対応しているのはAmazon S3のみ)

```
shell# 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
```

# リストアオプション: クラウドストレージとの直接連携

- Amazon S3上のバックアップをリストア  
(MEB3.11時点に対応しているのはAmazon S3のみ)

```
shell# 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
```

# バックアップオプション:暗号化

- 暗号化キーを直接指定する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --encrypt ¥  
> --key=23D987F3A047B475C900127148F9E0394857983645192874A2B3049570C12A34 ¥  
> --backup-dir=/var/tmp/backup backup-to-image
```

- ファイルに格納した暗号化キーを使用する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --encrypt ¥  
> --key-file=/meb/key --backup-dir=/var/tmp/backup backup-to-image
```

# バックアップオプション: 復号化

- ファイルに格納した復号化キーを直接指定する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --decrypt ¥  
> --key-file=/meb/key --backup-dir=/backups/extract-dir extract
```



# MySQL Enterprise Edition

# MySQL Enterprise Edition

ビジネス・クリティカルな環境において、最高レベルのMySQLスケーラビリティ、セキュリティ、信頼性、アップタイムを実現し、ビジネス・クリティカルな環境においてリスクとコストの削減を実現



MySQL導入の最適化



ROIの最適化をサポート



ユーザビリティ・顧客満足度の向上



# MySQL Enterprise Edition のサービスカテゴリ



## 拡張機能

- 拡張性
- 高可用性
- セキュリティ
- 監査
- 暗号化



## 管理ツール

- 監視
- バックアップ
- 開発
- 管理
- マイグレーション



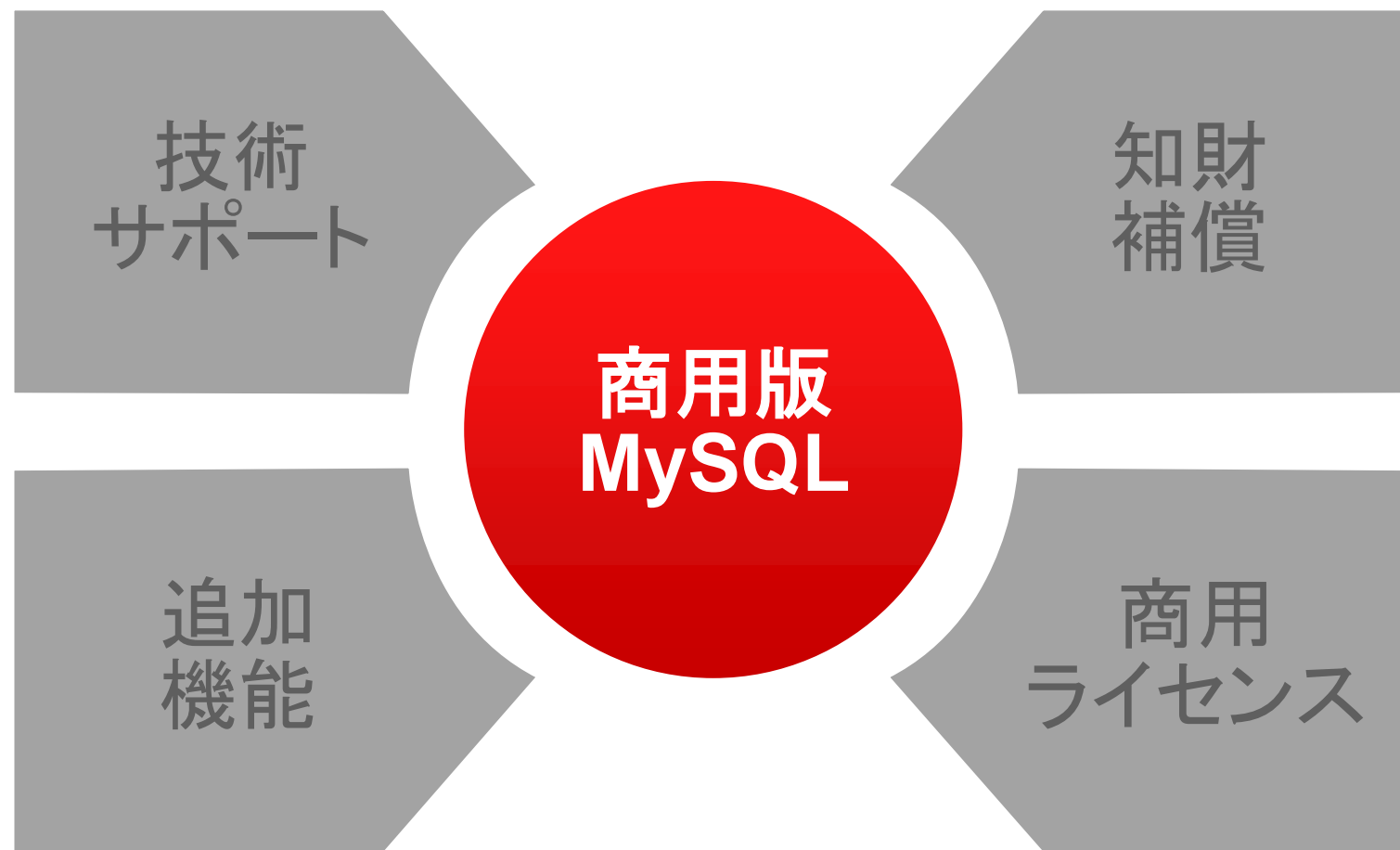
## サポート

- 技術サポート
- コンサルティングサポート
- オラクル製品との動作保証



# 商用版MySQLがご提供する価値

## 費用対効果の高い付加価値



	MySQL Editions		
	Standard Edition	Enterprise Edition	Cluster CGE
<b>機能概要</b>			
MySQL Database	✓	✓	✓
MySQL Connectors	✓	✓	✓
MySQL Replication	✓	✓	✓
MySQL Fabric		✓	✓
MySQL Partitioning		✓	✓
MySQL Utilities		✓	✓
Storage Engine: MyISAM, InnoDB	✓	✓	✓
Storage Engine: NDB (ndbcluster)			✓
MySQL Workbench SE/EE*	✓	✓	✓
MySQL Enterprise Monitor*		✓	✓
MySQL Enterprise Backup*		✓	✓
MySQL Enterprise Authentication (外部認証サポート) *		✓	✓
MySQL Enterprise Audit (ポリシーベース監査機能) *		✓	✓
MySQL Enterprise Encryption (非対称暗号化)*		✓	✓
MySQL Enterprise Firewall (SQLインジェクション対策)*		✓	✓
MySQL Enterprise Scalability (スレッドプール) *		✓	✓
MySQL Enterprise High Availability (HAサポート) *		✓	✓
Oracle Enterprise Manager for MySQL*		✓	✓
MySQL Cluster Manager (MySQL Cluster管理) *			✓
MySQL Cluster Geo-Replication			✓

\*商用版のみで利用可能な追加機能



	MySQL Editions		
	Standard SE	Enterprise EE	Cluster CGE
<b>Oracle Premium Support</b>			
24時間365日サポート	✓	✓	✓
インシデント数無制限	✓	✓	✓
ナレッジベース	✓	✓	✓
バグ修正&パッチ提供	✓	✓	✓
コンサルティングサポート	✓	✓	✓
<b>オラクル製品との動作保証</b>			
Oracle Linux	✓	✓	✓
Oracle VM	✓	✓	✓
Oracle Solaris	✓	✓	✓
Oracle Enterprise Manager		✓	✓
Oracle GoldenGate		✓	✓
Oracle Data Integrator		✓	✓
Oracle Fusion Middleware		✓	✓
Oracle Secure Backup		✓	✓
Oracle Audit Vault and Database Firewall		✓	✓

※最新の対比表は、[MySQL Editionsのサイト](#)を参照下さい。

# MySQL Supportの特徴

- 「パフォーマンス・チューニング」や「SQLチューニング」まで通常サポートの範囲内
  - コンサルティングサポートが含まれており、「クエリ・レビュー」、「パフォーマンス・チューニング」、「レプリケーション・レビュー」、「パーティショニング・レビュー」などに対応可能  
<http://www-jp.mysql.com/support/consultative.html>
- ソースコードレベルでサポート可能
  - ほとんどのサポートエンジニアがソースを読めるため、対応が早い
  - 開発エンジニアとサポートエンジニアも密に連携している
- 物理サーバー単位課金
  - CPU数、コア数に依存しない価格体系
- オラクルのライフタイムサポート
  - サポートポリシーが明確であるため、長期的な計画を立てやすい  
<http://www-jp.mysql.com/support/>

# 参考情報



# 参考情報

- MySQL Webサイト  
<https://www-jp.mysql.com/>
- MySQLコミュニティWebページ  
<http://dev.mysql.com/>
- 日本MySQLユーザー会(メーリングリスト有り)  
<http://www.mysql.gr.jp/>
- イベント案内
  - mysql.comのイベントページ  
<https://www-jp.mysql.com/news-and-events/events/>
  - オラクル社全体のイベントページ(OTN Japan - イベント・セミナー)  
<http://events.oracle.com/search/search>

# MySQLのドキュメント

- MySQL Developer Zone(<http://dev.mysql.com/>)にドキュメント類が公開されている
- 以下のドキュメントは2015年6月に日本語版が公開された
  - MySQL 5.6 リファレンスマニュアル (含むMySQL Cluster 7.3-7.4 マニュアル)  
<http://dev.mysql.com/doc/refman/5.6/ja/index.html>
  - MySQL Enterprise Monitor 3.0.18 マニュアル  
<http://dev.mysql.com/doc/mysql-monitor/3.0/ja/index.html>
  - **MySQL Enterprise Backup ユーザーズガイド (バージョン 3.11.1)**  
<http://dev.mysql.com/doc/mysql-enterprise-backup/3.11/ja/index.html>
- 上記日本語版公開以降に英語版ドキュメントのみ修正されている内容もあるため、ドキュメント参照時は英語版ドキュメントも合わせてご参照下さい。  
(URLの"ja"部分を"en"に変更すると、英語版ドキュメントが表示可能)

# MySQLのドキュメント

- MySQL Documentation: MySQL Reference Manuals  
<http://dev.mysql.com/doc/>
- MySQL Documentation: MySQL Workbench  
<http://dev.mysql.com/doc/index-gui.html>
- MySQL Documentation: MySQL Utilities/MySQL Fabric  
<http://dev.mysql.com/doc/index-utils-fabric.html>
- MySQL Documentation: Connectors and APIs  
<http://dev.mysql.com/doc/index-connectors.html>

# MySQLのドキュメント

- MySQL Documentation: Other MySQL Documentation  
<http://dev.mysql.com/doc/index-other.html>  
⇒ "world database"などのサンプルデータベースもダウンロード可能
- MySQL Documentation: MySQL Enterprise Products  
<http://dev.mysql.com/doc/index-enterprise.html>  
⇒ 商用版製品に関するドキュメント

# MySQL Enterprise Edition & Cluster CGEの試使用

## 30日間トライアル

The screenshot shows the Oracle Software Delivery Cloud interface. At the top, there's a blue header with the Oracle logo and 'Software Delivery Cloud'. Below it, a progress bar indicates three steps: '条件および規制' (Conditions and Restrictions), '検索' (Search), and 'ダウンロード' (Download). The '検索' step is currently active. The main content area is titled 'メディア・パック検索' (Media Pack Search) and includes a '手順' (Procedure) section with three numbered steps. Below the steps, there are two dropdown menus: '製品パックを選択' (Select Product Pack) set to 'MySQL Database' and 'プラットフォーム' (Platform) set to 'Linux x86-64'. An '実行' (Execute) button is visible. Below the search area, a '結果' (Results) section shows a table header with columns: '選択' (Select), '説明' (Description), 'リリース' (Release), '部品番号' (Part Number), '更新' (Update), and '部品数 / サイズ' (Part Count / Size). A message states '\*\*\* 検索はまだ実行されていません \*\*\*' (\*\*\* Search has not yet been executed \*\*\*). A blue arrow points down to a table of search results.

選択	説明	リリース	部品番号	更新	部品数 / サイズ
	*** 検索はまだ実行されていません ***				
ダウンロード	MySQL Cluster 7.2.4 TAR for Generic Linux 2.6 x86 (64bit)		V30623-01		301M
ダウンロード	MySQL Cluster Manager 1.1.4+Cluster for Red Hat and Oracle Linux 5 x86 (64-bit)		V30517-01		257M
ダウンロード	MySQL Cluster Manager 1.1.4+Cluster for SuSE Enterprise Linux 11 x86 (64-bit)		V30519-01		257M
ダウンロード	MySQL Cluster Manager 1.1.4+Cluster for SuSE Enterprise Linux 10 x86 (64-bit)		V30518-01		257M
ダウンロード	MySQL Cluster Manager 1.1.4 for Red Hat and Oracle Linux 5 x86 (64-bit)		V30492-01		13M

• Oracle Software Delivery Cloud  
<http://edelivery.oracle.com/>

• 製品パックを選択：  
“MySQL Database”

• 製品マニュアル  
<http://dev.mysql.com/doc/index-enterprise.html>

# オラクルユニバーシティ MySQL 研修

コース名	日数	価格 (税込)	開催日程
MySQL for Beginners	4	¥220,320	お問い合わせください
MySQL データベース管理 I	3	¥165,240	2015/09/07 - 09, 2015/10/26 - 28, 2015/12/02 - 04
MySQL データベース管理 II	2	¥110,160	2015/09/24 - 25, 2015/10/05 - 06, 2015/09/29 - 30, 2015/12/07 - 08
MySQL High Availability	3	¥231,336	お問い合わせください

※ MySQL データベース管理 I/II, MySQL Performance TuningはMySQL5.5対応、MySQL 入門は MySQL 5.0/5.1対応です。

※ コース開催予定は2015年7月現在のもので、開催日程の最新情報はOracle University ホームページ (<http://www.oracle.com/jp/education/>)にてご確認ください。

※ 価格(税込み)は**2015年7月現在**の価格です。Oracle PartnerNetwork 会員様は、パートナー割引価格で受講いただけます。

# 管理者向け MySQL 5.6 対応認定資格

- Oracle Certified **Professional**, MySQL **5.6** Database Administrator
  - Oracle Certified Professional, MySQL 5.6 Database Administrator 資格は、パーティショニング、およびレプリケーションにおける機能強化やパフォーマンス監視と診断のPERFORMANCE\_SCHEMAの使用などMySQL 5.6の新機能を含むMySQLデータベースのインストール、複製、チューニング、およびセキュリティ設定など幅広い管理スキルを証明します。
- 認定試験:
  - MySQL 5.6 Database Administrator (1Z0-883)
    - 本試験に合格することで、資格取得できます
    - 日本語試験、英語試験共に受験可能

# 開発者向け MySQL 5.6 対応認定資格

- Oracle Certified **Professional**, MySQL **5.6** Developer
  - Oracle Certified Professional, MySQL 5.6 Database Administrator 資格は、MySQL データ・タイプや SQL シンタックス、テーブルやスキーマなどの各種オブジェクト、ストアド・プロシージャ、ビュー、結合など、MySQL データベースを使用したアプリケーション開発に必要なスキルを証明します。
- 認定試験:
  - MySQL 5.6 Developer (1Z0-882)
    - 本試験に合格することで、資格取得できます
    - 日本語試験、英語試験共に受験可能



# 管理者向け MySQL 5.6 認定資格取得パス

新規取得もアップグレードも一試験で。

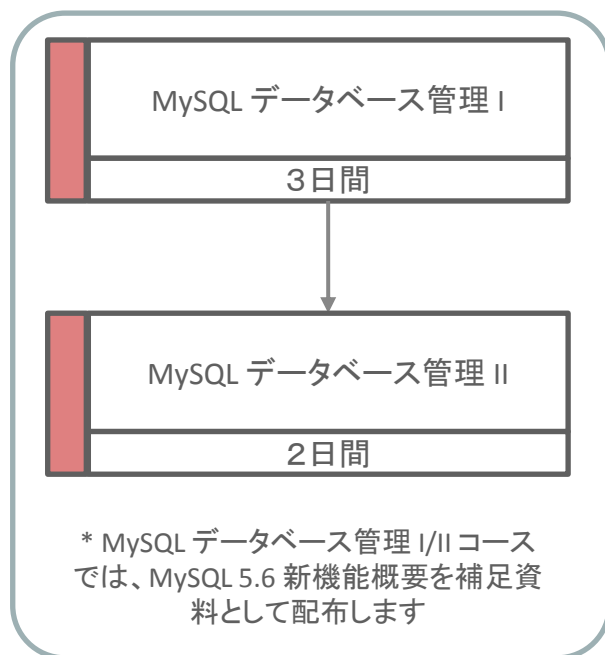
学習(研修受講)

受験

資格取得



これから  
資格取得を  
目指す方



1Z0-883:  
MySQL 5.6  
Database Administrator



OCP MySQL  
5 DBA  
資格取得者

Oracle Certified **Professional**,  
MySQL **5.6** Database Administrator

→ 必須

- - - - - 推奨

# **Hardware and Software Engineered to Work Together**

ORACLE®