



DevOps Practical Test - Task: Bash Scripting for backups

1. Overview

The MySQL Backup Script automates the process of creating compressed backups of MySQL databases. It is designed for local backups and provides flexible configuration through command-line arguments. The script ensures secure password entry and includes functionality for backup retention management and cleanup.

2. Key Features

- **Secure Password Prompt:** Password entry is secured using `read -s`, ensuring the password is not displayed on the terminal.
- **Mandatory Command-Line Arguments:** Requires specific arguments (`db_name`, `db_user`, `backup_dir`) to be provided, ensuring essential information is always specified.
- **Automatic Installation of `mysqldump`:** Checks for the presence of `mysqldump` and installs it if not found, based on the system's package manager.
- **Backup Retention Management:** Optionally cleans up old backups, retaining only those within a specified number of days.

3. Script Workflow

1. **Check for `mysqldump` Installation:**
 - Verifies if `mysqldump` is installed. If not, attempts to install it using the system's package manager.
2. **Secure Password Prompt:**
 - Prompts the user for the MySQL password securely using `read -s`.
3. **Parse Command-Line Arguments:**
 - Processes command-line arguments for database configuration and backup options. Ensures mandatory arguments are provided.
4. **Create Backup:**
 - Uses `mysqldump` to generate a compressed backup of the specified MySQL database and stores it in the specified local directory.
5. **Cleanup Old Backups:**

- If the cleanup option is specified, deletes backup files older than the specified retention period.

4. Command-Line Arguments

Option	Description	Default Value
<code>-d db_name</code>	Mandatory: Database name to back up	N/A
<code>-u db_user</code>	Mandatory: Database user for authentication	N/A
<code>-b backup_dir</code>	Mandatory: Local directory to store backups	N/A
<code>-h db_host</code>	Database host	<code>localhost</code>
<code>-P db_port</code>	Database port	<code>3306</code>
<code>-R retention_days</code>	Number of days to keep backups	<code>7</code>
<code>-c</code>	Perform cleanup of old backups	N/A

5. Usage Examples

Script manual page

```
man ./mysql_backup.1
```

```
>> ~/D/w/2/mysql_backup_script on feature/mysql-backup-script x man ./mysql_backup.1
>> ~/D/w/2/mysql_backup_script on feature/mysql-backup-script x █
```

The command man will print the manual of the script:

```
NAME
    mysql_backup - A script to backup MySQL databases for WIS DevOps Practical Test

SYNOPSIS
    mysql_backup [OPTIONS]

DESCRIPTION
    This script creates a compressed backup of a MySQL database and stores it in a local directory.

    The script supports command-line arguments to configure the database connection, backup location, and retention period.

OPTIONS
    -d db_name
        Specify the database name (mandatory).

    -u db_user
        Specify the database user (mandatory).

    -b backup_dir
        Specify the local backup directory (mandatory).

    -h db_host
        Specify the database host (default: localhost).
```

Basic Backup Command:

```
./mysql_backup.sh -d my_database -u my_user -b /my/backup/dir
```

Backs up `my_database` to `/my/backup/dir`. Password will be prompted.

Backup with Custom Retention and Cleanup:

```
./backup_script.sh -d my_database -u my_user -b /my/backup/dir -R 14
-c
```

- Backs up `my_database` and cleans up backups older than 14 days.

6. Error Handling

- **Missing Mandatory Arguments:** The script exits with an error message if `

`db_name`, `db_user`, or `backup_dir` is missing from the command-line arguments, displaying usage information to guide the user.

7. Script Limitations

- **Local Backup Only:** The script is designed for local backups and does not include functionality for remote backup transfers.
- **OS-Specific Installation:** The installation process for `mysqldump` depends on the availability of a compatible package manager and might not be supported on all operating systems.

8. Future Enhancements

- **Remote Backup Support:** Integrate options for transferring backups to a remote server.
- **Enhanced Error Handling:** Implement more robust error checking for database connectivity and file operations.
- **Multiple Database Support:** Extend functionality to support backing up multiple databases in one execution.