MySQL Automation - 2 node down in 3 node cluster

Overview

This Python script is designed to monitor and manage MySQL database servers in two clusters. It performs ping checks, connects to servers, handles server promotion and demotion, and logs events for failover and failback operations.

Dependencies

- `subprocess`: To execute shell commands such as `ping`.
- `time`: For sleep intervals between checks.
- `socket`: To resolve IP addresses to hostnames.
- `mysql.connector`: To connect to MySQL servers.
- `os`: For directory and file operations.

User Inputs

- **Production Database Server IP**: IP address of the primary production database server.
- **High Availability Database IP**: IP address of the high availability (HA) server.
- **Near DR Database IP**: IP address of the near disaster recovery (DR) server.
- **DR Database IP**: IP address of the disaster recovery (DR) server.

Global Variables

- `database_cluster_1`: List containing IPs for the production, high availability, and near DR servers.
- `database_cluster_2`: List containing the IP for the DR server.
- `check_interval`: Time interval (in seconds) between checks.

- `log_file_path`: Path to the log file.
- `current_primary_file`: Path to the file tracking the current primary server.

Functions

`ensure_log_folder_exists()`

- **Purpose**: Creates the log directory if it does not exist.
- **Returns**: None.

`log_event(message)`

- **Purpose**: Logs a message with a timestamp to the log file and optionally prints it to the console.
- **Parameters**:
  - `message` (string): Message to be logged.
- **Returns**: None.

`get_hostname_ip(ip_address)`

- **Purpose**: Retrieves the hostname for a given IP address.
- **Parameters**:
  - `ip_address` (string): IP address to resolve.
- **Returns**: Hostname or error message.

`connect_mysql_database_server(host, port, user, password)`

- **Purpose**: Connects to a MySQL server and logs the connection status.
- **Parameters**:
  - `host` (string): Hostname or IP address of the MySQL server.
  - `port` (int): Port number of the MySQL server.
  - `user` (string): Username for MySQL authentication.
  - `password` (string): Password for MySQL authentication.
- **Returns**: MySQL connection object if successful; otherwise, `None`.

`ping_database_cluster(server)`

- **Purpose**: Pings a server to check its availability.
- **Parameters**:
  - `server` (string): IP address of the server to ping.
- **Returns**: A tuple `(success, output)` where `success` is a boolean indicating if the ping was successful and `output` contains the command output.

`promotion_commands(connection)`

- **Purpose**: Executes commands to promote a server to primary.
- **Parameters**:
  - `connection` (MySQL connection object): Connection to the MySQL server.
- **Returns**: None.

`demotion_commands(connection)`

- **Purpose**: Executes commands to demote a server to secondary.
- **Parameters**:
  - `connection` (MySQL connection object): Connection to the MySQL server.
- **Returns**: None.

`failover_to_secondary(server_ip)`

- **Purpose**: Demotes a server to secondary.
- **Parameters**:
  - `server_ip` (string): IP address of the server to demote.
- **Returns**: None.

`failback_to_primary(server_ip)`

- **Purpose**: Promotes a server to primary.
- **Parameters**:
  - `server_ip` (string): IP address of the server to promote.

- **Returns**: None.

`track_current_primary()`

- **Purpose**: Reads the IP address of the current primary server from the file.
- **Returns**: The IP address of the current primary server or `None` if the file does not exist.

`update_current_primary(ip)`

- **Purpose**: Updates the file with the new primary server IP address.
- **Parameters**:
  - `ip` (string): IP address of the new primary server.
- **Returns**: None.

`main_1()`

- **Purpose**: Monitors and manages Database Cluster 1. If all servers are down, it demotes the production and near DR servers and promotes the DR server. If any server is up, it performs related actions.
- **Returns**: None.

`main_2()`

- **Purpose**: Monitors and manages Database Cluster 2. If the DR server is down, it promotes production and near DR servers and demotes the DR server. If the DR server is up, it performs related actions.
- **Returns**: None.

Execution Loop

- The script continuously runs `main_1()` and `main_2()` in a loop, with a sleep interval defined by `check_interval`.
