- To use a global configuration file only, the file contents look like this:

```
{
  "path": "/usr/local/mysql/keyring/component_keyring_encrypted_file",
  "password": "password",
  "read_only": false
}
```

  Create this file in the directory where the `component_keyring_encrypted_file` library file is installed.

- Alternatively, to use a global and local configuration file pair, the global file looks like this:

```
{
  "read_local_config": true
}
```

  Create this file in the directory where the `component_keyring_encrypted_file` library file is installed.

  The local file looks like this:

```
{
  "path": "/usr/local/mysql/keyring/component_keyring_encrypted_file",
  "password": "password",
  "read_only": false
}
```

  Create this file in the data directory.

Keyring operations are transactional: `component_keyring_encrypted_file` uses a backup file during write operations to ensure that it can roll back to the original file if an operation fails. The backup file has the same name as the data file with a suffix of `.backup`.

`component_keyring_encrypted_file` supports the functions that comprise the standard MySQL Keyring service interface. Keyring operations performed by those functions are accessible at two levels:

- SQL interface: In SQL statements, call the functions described in Section 6.4.4.14, “General-Purpose Keyring Key-Management Functions”.
- C interface: In C-language code, call the keyring service functions described in Section 5.6.9.2, “The Keyring Service”.

Example (using the SQL interface):

```
SELECT keyring_key_generate('MyKey', 'AES', 32);
SELECT keyring_key_remove('MyKey');
```

For information about the characteristics of key values permitted by `component_keyring_encrypted_file`, see Section 6.4.4.12, “Supported Keyring Key Types and Lengths”.

6.4.4.6 Using the keyring_file File-Based Keyring Plugin

The `keyring_file` keyring plugin stores keyring data in a file local to the server host.

Warning

For encryption key management, the `keyring_file` plugin is not intended as a regulatory compliance solution. Security standards such as PCI, FIPS, and others