CASSANDRA-900

tzz

2010-03-15

Contents

1 Root issue CASSANDRA-900 **2**
1.1 Summary 2
1.2 Description 2
1.3 Commits 2
1.4 Comments 2
1.5 Pull requests 3

Chapter 1

Root issue CASSANDRA-900

1.1 Summary

access levels for Thrift authorization

1.2 Description

Provide access levels at the API level, set by the login() method relayed through IAuthenticator.

1.3 Commits

No related commits

1.4 Comments

1. **tzz:** This is a simple patch that will just replace checkLoginDone() with the appropriate access level check throughout CassandraServer. ZERO, READ, INSERT, and ALL access levels are proposed.

2. **tzz:** Passes all tests.

3. **tzz:** SimpleAuthenticator will simply grant ALL access. As discussed in the mailing lists, it won't try to be a comprehensive solution and users should implement the IAuthenticator that makes sense for them.

4. **urandom:** Hey Ted, I have a couple of questions regarding this.

   * Can you explain your choice of access levels? In particular, I"m not sure I understand why you'd want both INSERT and ALL, (particularly since the only difference is deletes which you can effectively do through an overwrite).

   * Can you explain why you'd want to return an AccessLevel to the client, and why you wouldn't throw the AuthorizationException (assuming that was by choice).

5. **tzz:** I separated an INSERT level for writers that shouldn't be able to delete (logging agents). Overwriting is not the same as deleting: you can only overwrite what you know; deleting can use ranges. This is a necessary use case in my environment.

I considered a DELETE access level too, since as you see INSERT and DELETE are really separate. Perhaps with a separate DELETE, the ALL AccessLevel won't be needed (see below) because it's an OR of READ+INSERT+DELETE.

The client gets back an AccessLevel so they know in advance what they've been authorized to do. Throwing an exception later in the game is still done. It's a single-byte return code, we already store it, and it won't change for the duration of the connection. I don't see the harm in sending it back. The client can just ignore it if they want. We could change the return to an int, though, so we can express "INSERT+DELETE" or "READ+INSERT" numerically without more AccessLevels.

6. **urandom:** I've applied this with some minor changes. Basically I changed the enum members to NONE, READONLY, READWRITE, and FULL in the hope that they better communicate their effect.

   Thanks Ted.

7. **hudson:** Integrated in Cassandra #388 (See [http://hudson.zones.apache.org/hudson/job/Cassandra/388/]) access levels for Thrift authorization

   Patch by Ted Zlatanov and eevans for
   regenerated thrift code to include new enum

   Patch by eevans and Ted Zlatanov for

8. **tzz:** Thanks for working on this, Eric.

   I forgot to change the API version in cassandra.thrift. This change should probably have bumped the minor rev. WDYT?

9. **urandom:** Actually, since the return type for login() has changed, I'd call that "backward incompatible" and say that the major needs to be incremented.

   Good catch, I completely missed this. I'll update it presently.

   Thanks Ted.

1.5 Pull requests

No pull requests