Item 222
git_comments:

1. * * String representation of this ZooKeeperAdmin client. Suitable for things * like logging. * * Do NOT count on the format of this string, it may change without * warning. * * @since 3.5.3
2. * * Create a ZooKeeperAdmin object which is used to perform dynamic reconfiguration * operations. * * @param connectString * comma separated host:port pairs, each corresponding to a zk * server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002" If * the optional chroot suffix is used the example would look * like: "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002/app/a" * where the client would be rooted at "/app/a" and all paths * would be relative to this root - ie getting/setting/etc... * "/foo/bar" would result in operations being run on * "/app/a/foo/bar" (from the server perspective). * @param sessionTimeout * session timeout in milliseconds * @param watcher * a watcher object which will be notified of state changes, may * also be notified for node events * @param conf * passing this conf object gives each client the flexibility of * configuring properties differently compared to other instances * * @throws IOException * in cases of network failure * @throws IllegalArgumentException * if an invalid chroot path is specified * * @see ZooKeeper#ZooKeeper(String, int, Watcher, ZKClientConfig)
3. * * The Asynchronous version of reconfig. * * @see #reconfig * *
4. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
5. * * Create a ZooKeeperAdmin object which is used to perform dynamic reconfiguration * operations. * * @param connectString * comma separated host:port pairs, each corresponding to a zk * server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002" If * the optional chroot suffix is used the example would look * like: "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002/app/a" * where the client would be rooted at "/app/a" and all paths * would be relative to this root - ie getting/setting/etc... * "/foo/bar" would result in operations being run on * "/app/a/foo/bar" (from the server perspective). * @param sessionTimeout * session timeout in milliseconds * @param watcher * a watcher object which will be notified of state changes, may * also be notified for node events * * @throws IOException * in cases of network failure * @throws IllegalArgumentException * if an invalid chroot path is specified * * @see ZooKeeper#ZooKeeper(String, int, Watcher) *
6. * * Reconfigure - add/remove servers. Return the new configuration. * @param joiningServers * a comma separated list of servers being added (incremental reconfiguration) * @param leavingServers * a comma separated list of servers being removed (incremental reconfiguration) * @param newMembers * a comma separated list of new membership (non-incremental reconfiguration) * @param fromConfig * version of the current configuration * (optional - causes reconfiguration to throw an exception if configuration is no longer current) * @param stat the stat of /zookeeper/config znode will be copied to this * parameter if not null. * @return new configuration * @throws InterruptedException If the server transaction is interrupted. * @throws KeeperException If the server signals an error with a non-zero error code.
7. * * Create a ZooKeeperAdmin object which is used to perform dynamic reconfiguration * operations. * * @param connectString * comma separated host:port pairs, each corresponding to a zk * server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002" If * the optional chroot suffix is used the example would look * like: "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002/app/a" * where the client would be rooted at "/app/a" and all paths * would be relative to this root - ie getting/setting/etc... * "/foo/bar" would result in operations being run on * "/app/a/foo/bar" (from the server perspective). * @param sessionTimeout * session timeout in milliseconds * @param watcher * a watcher object which will be notified of state changes, may * also be notified for node events * @param canBeReadOnly * whether the created client is allowed to go to * read-only mode in case of partitioning. Read-only mode * basically means that if the client can't find any majority * servers but there's partitioned server it could reach, it * connects to one in read-only mode, i.e. read requests are * allowed while write requests are not. It continues seeking for * majority in the background. * * @throws IOException * in cases of network failure * @throws IllegalArgumentException * if an invalid chroot path is specified * * @see ZooKeeper#ZooKeeper(String, int, Watcher, boolean)
8. * * This is the main class for ZooKeeperAdmin client library. * This library is used to perform cluster administration tasks, * such as reconfigure cluster membership. The ZooKeeperAdmin class * inherits ZooKeeper and has similar usage pattern as ZooKeeper class. * Please check {@link ZooKeeper} class document for more details. * * @since 3.5.3
9. Now enable reconfig feature by turning on the switch.
10. election port
11. Use DigestAuthenticationProvider.base64Encode or run ZooKeeper jar with org.apache.zookeeper.server.auth.DigestAuthenticationProvider to generate password. An example: java -cp zookeeper-3.6.0-SNAPSHOT.jar:lib/log4j-1.2.17.jar:lib/slf4j-log4j12-1.7.5.jar: lib/slf4j-api-1.7.5.jar org.apache.zookeeper.server.auth.DigestAuthenticationProvider super:test The password here is 'test'.
12. Ignore.
13. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
14. password is test
15. new client port
16. However a failure is still expected as user is not authenticated, so ACL check will fail.
17. Get a three server quorum.
18. There is ACL however the permission is wrong - need WRITE permission at leaste.
19. quorum port
20. Utility method that recreates a new ZooKeeperAdmin handle, and wait for the handle to connect to quorum servers.

21. Again failure is expected because no ACL is associated with this user.
22. election port
23. Ignore.
24. This tests the case where ZK ensemble does not have the super user's password configured. Reconfig should fail as the super user has to be explicitly configured via zookeeper.DigestAuthenticationProvider.superDigest.
25. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
26. new client port
27. Get a three server quorum.
28. quorum port
29. !< Attempts to perform a reconfiguration operation when reconfiguration feature is disabled
30. remove the follower.
31. * * ZOOKEEPER-2014: only admin or users who are explicitly granted permission can do reconfig.
32. All cases should fail as server ensemble was not configured with the super user's password.
33. password is test
34. No auth, should fail.
35. Right auth, should pass.
36. Create a new quorum with the super user's password not configured.
37. Wrong auth, should fail.
38. Wait until all the servers start, and fail if they don't start within 10 seconds.
39. The server hasn't started.
40. Append additional config, if any.
41. Additional environment variables when starting zkServer.sh.
42. Additional config options as a list of key/value pairs.
43. * Attempts to perform a reconfiguration operation when reconfiguration feature is disabled.
44. * * @see Code#RECONFIGDISABLED
45. This should never happen when executing reconfig command line, because it is guaranteed that we have a ZooKeeperAdmin instance ready to use in CliCommand stack. The only exception would be in test code where clients can directly set ZooKeeper object to ZooKeeperMain.
46. Reconfig node is access controlled by default (ZOOKEEPER-2014).
47. Need check if the record is a DataNode instance because of changes in ZOOKEEPER-2014 which adds default ACL to config node.
48. password is 'test'
49. password is 'test'
50. password is 'test'
51. password is 'test'
52. password is 'test'
53. server ids are 1, 2 and 3
54. create an extra handle, so we can index the handles from 1 to qu.ALL using the server id.
55. **comment:** not used.
    label: code-design
56. password is 'test'
57. password is 'test'

git_commits:

1. **summary:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
   message: ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/ Author: Michael Han <hanm@cloudera.com> Reviewers: fpj <fpj@apache.org>, breed <breed@apache.org>, rgs <rgs@itevenworks.net> Closes #96 from hanm/ZOOKEEPER-2014

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
   body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
2. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
   body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
3. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
   body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
4. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.

body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
label: code-design
5. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
6. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
7. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
8. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
9. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
label: code-design
10. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
11. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
label: code-design
12. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
label: code-design
13. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
14. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
15. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
16. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/
17. **title:** ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster.
body: This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/

github_pulls_comments:

1. Thanks @breed and @rgs1 for your time and review feedback. Pull request, patch, and review board is now updated.

github_pulls_reviews:

1. connect to follower seems repeated, can we move it to a private helper method?
2. **body:** nit: spaces around `!=`
label: code-design
3. i think this should be an assertion (and we can drop the LOG.error call)
4. depends -> depending
5. actually if we are asserting above, perhaps we should also assert here.
6. is there a reason we added the instanceof here? if we didn't need it before, why do we need it now?
7. **body:** this isn't needed anymore right?
label: code-design
8. we don't throw NoNodeException anymore
9. **body:** +1 awesome work! i only found little nits! thanx for sticking with this!
label: code-design
10. **body:** Agreed, both places are updated to use asserts and removed logging.
label: code-design
11. I have to double check the code again, but as far as I remember, we did not need this type check previously because a data tree that's created by ZooKeeper itself will only contain znode (DataNode) record. Now with this patch, we implicitly create ACL that appertains to /zookeeper/config node while creating a DataTree, so when serializing DataTree a record could be an ACL instead of a DataNode.
12. Good point, updated patch.
13. @hanm hmm, I'm not sure about this. In the changes for `DataTree`, we only set the ACL of the `/zookeeper/config` znode, but setting ACLs was something we were doing before, so I'm confused about why we can have a mix of znode records and ACL

records with the changes proposed here. Could you clarify, please?

14. @fpj - we were not setting ACLs on intrinsic znodes (i.e. /zookeeper/config) ZooKeeper implicitly created while initializing a DataTree before. And for this test case, it only creates znodes, not ACLs. As a result, it's reasonable for the previous test case to assume every record that's serializing is a DataNode record. Now with this patch, there is an ACL implicitly created when /zookeeper/config node is created, so the previous assumption (that all records to be serialized are DataNode record) does not hold. Thus, a change is required. For reference, you could put a break point on https://github.com/apache/zookeeper/blob/master/src/java/main/org/apache/zookeeper/server/ReferenceCountedACLCache.java#L133 while running this test case, and you will see there is one ACL that's serialized. Now you can remove the ACL associated with /zookeeper/config at https://github.com/apache/zookeeper/pull/96/files#diff-a676d93082759105dd8c79c0a76a8007R259, and you will see the break point on ReferenceCountedACLCache.java previous set not get hit. That is the difference. Another way to experiment this is to create an ACL in this test (without applying this pull request first), something like: `final DataNode markerNode = tree.getNode("/marker"); tree.setACL("/marker", ZooDefs.Ids.READ_ACL_UNSAFE, -1);` will do. Then we will see the same type casting failure - this simulates what this PR will do in terms of changing the type of records. Basically I think the root cause is the test itself could be made more robust, by eliminate the assumptions (that every record is a DataNode) that might not always hold.

jira_issues:

1. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
2. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
3. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
4. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
5. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
6. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
7. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
8. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have

to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/

9. **summary:** Only admin should be allowed to reconfig a cluster
   description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
10. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
11. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
    label: code-design
12. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
13. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
14. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
15. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
16. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
17. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
18. **summary:** Only admin should be allowed to reconfig a cluster
    description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/

label: documentation

19. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
20. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
21. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
22. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
23. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design
24. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
25. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
26. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
27. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
28. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
29. **summary:** Only admin should be allowed to reconfig a cluster

description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/

30. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
31. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
32. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
33. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
34. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
35. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
36. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
37. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
38. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
39. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also

force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: test

40. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
41. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
42. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
43. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
44. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design
45. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
46. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
47. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: documentation
48. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
49. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also

force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design

50. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
51. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: test
52. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
53. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
54. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
55. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
56. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
57. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
58. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
59. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have

to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design

60. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
61. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
62. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design
63. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design
64. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
65. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
66. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
67. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
68. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
69. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have

to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/

70. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
71. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
label: code-design
72. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
73. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/
74. **summary:** Only admin should be allowed to reconfig a cluster
description: ZOOKEEPER-107 introduces reconfiguration support via the reconfig() call. We should, at the very least, ensure that only the Admin can reconfigure a cluster. Perhaps restricting access to /zookeeper/config as well, though this is debatable. Surely one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. We could also force a default ACL to make it a bit more consistent (maybe). Finally, making reconfig() only available to Admins means they have to run with zookeeper.DigestAuthenticationProvider.superDigest (which I am not sure if everyone does, or how would it work with other authentication providers). Review board https://reviews.apache.org/r/51546/

jira_issues_comments:

1. this is RFC and very basic. cc: [~shralex], [~phunt], [~hdeng]
2. {quote} ensure that only the Admin can reconfigure a cluster. {quote} This change will reduce much flexibility in reconfig. A counter scenario would be I have a process that detects "permanent" failed ZK servers and removes them to make a smaller quorum (better fault tolerance). Does the process have to be Admin? Or would a default ACL be a better option here?
3. Thanks for starting this thread, Raul. I'm not sure what's the right way to handle the issue, but just wanted to mention a couple of things. 1. Regarding access to /zookeeper/config - clients may need read access for example in order to run the new client-side load balancing functionality, they need to detect a config change and get a new connection string. They can use getConfig for this. It is probably a good idea to restrict access for writes. Writing it probably won't break the system but may cause all clients to migrate to any server I choose to mention there, if clients are using the client-side load balancing algorithm. 2. Notice that when the leader processes reconfig in PrepRequestProcessor it already checks ACL: nodeRecord = getRecordForPath(ZooDefs.CONFIG_NODE); checkACL(zks, nodeRecord.acl, ZooDefs.Perms.WRITE, request.authInfo);
4. Actually perhaps we should open a JIRA to hide the client-side rebalancing from clients (not for 3.5.0). We may want to implement it inside the client-side library somehow and just let the client specify whether this is enabled or disabled when creating a ZK handle. What do you think ?
5. +[~fpj], [~michim]
6. [~rgs], should this be similar to how updating the quota znode works ? or do you think changing configuration is different ?
7. [~shralex]: well, quotas rely on ACLs. But, quotas are (currently) advisory only (i.e.: we don't reject traffic) so no damage can be done by non-admins. So I think we need a stricter enforcement here, no?
8. (by which I mean, you can really bring a cluster down with this).
9. Yes, this is a concern for me as well. 1) mixing the client api and the admin api is not great. It would be better to have them separate. We should fix this asap. 2) this (controlling access to reconfig) is a big issue from a security perspective IMO. A few comments on the comments so far: bq. ensure that only the Admin can reconfigure a cluster sounds sensible to me bq. Perhaps restricting access to /zookeeper/config as well in the past we've (ben in particular) tried to limit the amount of information we provide to the client/session. For example we don't tell them which server they are connected to. I see this in the same vein. bq. one could ensure Admin only access via an ACL, but that would leave everyone who doesn't use ACLs unprotected. well, you're already unprotected in this situation so I don't really see it as a sticking point. bq. clients may need read access for example in order to run the new client-side load balancing functionality perhaps this argues for pushing this to the the server? encapsulate the information on the service I mean and expose as a specific api. bq. Actually perhaps we should open a JIRA to hide the client-side rebalancing from clients (not for 3.5.0). yes, that's what I was trying to get at in the previous item in this comment. Although I think we'd need to fix this now - while we can still change the apis w/o worrying about b/w compat (we can change new apis during the alpha period w/o worrying about back compat) bq. should this be similar to how updating the quota znode works ? or do you think changing configuration is different ? if quotas are "admin" functions we probably need to fix those as well - lock them down to just admin level authz I mean.

10. Everything you guys are saying about admin-only controls sounds very reasonable. I just want to clarify about the special reconfig znode. IMHO we should not allow write permissions to this node. I don't even see why an admin should have it :) its set only through the reconfig API. I do think that all clients should have read permissions, and here's why - the information they get from this znode is the up-to-date connection string. When the configuration changes this is the bare minimum they need in order not to loose track of the system. When their server crashes they need to know whom to connect to next. The new connection string is exactly the information we exploit for rebalancing. It is even implemented inside the updateServerList method, which is also needed in any case. Regarding my suggestion - Hongchao opened a JIRA for it and you can read the discussion there https://issues.apache.org/jira/browse/ZOOKEEPER-2016 Please especially see Marshall's comment. All I'm proposing there is to implement some default behaviour that will save most clients from setting a watch and invoking updateServerList - I suggest that the client-side-library does it for them if they opt-in for the default behaviour. It doesn't change APIs, just adds one more feature, so it doesn't delay 3.5.0.
11. **body:** [~rgs], does your patch solve the issue ? if not, what is still missing ? If I remember correctly my concern was that I'd like getConfig to be available to regular clients, not only admin, so they can react to configuration changes. If this JIRA is what's blocking 3.5 perhaps we could reconsider the approach and go with something simpler to start with, such as relying on ACLs. Or setting default ACLs for the config znode and requiring client admins to have these permissions.
    label: code-design
12. We can't (or at least promise we won't) change the api in a non b/w compatible way once it's post-alpha. As such I believe at a minimum we need to clean up the API (should be simple - move it out of ZooKeeper class) and ensure there are no security issues.
13. Had an offline discussion with [~phunt], [~fpj] yesterday regarding this issue, the conversation is captured as following points (with some of my thoughts as well): * We can't fix security issue unless we enforce authentication and authorization. Just by moving client APIs around is not enough because at protocol level the server still open to reconfiguration and someone can exploit this easily (by writing their own ZK client instead of using ZooKeeper client for example.). * That said though, the ZooKeeper::reconfig API should be moved out of ZooKeeper class (for Java client) anyway, because it's more about an admin feature rather than a client API. Moving reconfig out of ZooKeeper will also remove constraints we possibly put on normal ZooKeeper clients (such as having to use zookeeper.DigestAuthenticationProvider.superDigest), which is a bonus. Due to API backward compatibility concerns, this refactoring should happen before we move to beta. * We'd like to introduce a new configuration option in zoo.cfg to turn off reconfig feature by default. ZK users who needs use this feature need turn it on explicitly. Because dynamic reconfig feature brings something new (e.g. cfg.dynamic file), have this feature off is good for 'the principal of least surprise'. Having the feature off by default will also buy us sometime to fortify and stabilize the feature without having it being a blocker issue. The action items / plans I am thinking in the time frame of 3.5.3: * Fix Security: #* Enforce an ACL on /zookeeper/config, such that only users that have write permission to it can reconfig the cluster. /zookeeper/config is by default readable to anyone so zookeeper (none-admin) clients can load balancing on client side either manually (current behavior) or automatically (ZOOKEEPER-2016). #* ZooKeeper users are responsible for properly configure the ACL such that only a limited set of admin users are part of the ACL with write access. The authentication of these users will be delegated to existing mechanisms ZooKeeper already supports, such as SASL client login, so not much work here except documentation on such a requirement to use reconfig feature. #* The default behavior is such that if /zookeeper/config node has no associated ACLs, then no one is allowed write access except super user. * Fix API: #* For Java client, create a new class ZooKeeperAdmin that inherits ZooKeeper class and move reconfig API into ZooKeeperAdmin class. #* For C client, there is no namespace and the existing zhandle is coupled with reconfig state so it is not as simple as Java client to isolate admin part of the API from rest of APIs. Working on a proposal. * Introduce a new zoo.cfg option to switch reconfiguration feature on or off. The new configuration option will be set as 'disable reconfig' by default when 3.5.3 shipped, and user who wants reconfig has to enable reconfig option explicitly. Are these good enough for now to address all security concerns or we still miss something? Comments? CC [~shralex], [~rgs]
14. Hi Michael, all of this sounds good to me except for moving getConfig to ZooKeeperAdmin. I've explained why several times in the thread above (search for "getconfig"), please let me know if you disagree.
15. Thanks for your feedback Alex. I agree about getConfig API remaining as part of ZooKeeper because we don't transparently handle load balancing for clients. I had that in my first version of proposal but I forgot to keep it when update the proposal. A side question, I think this (handle load balancing when server changes) is the only use case for getConfig API right? Basically: * Client sets a watcher when calling getConfig. * The callback of the watcher invokes updateServerList. * When servers are reconfigured, watcher gets triggered and updateServerList gets invoked, which would do load balancing for clients.
16. I think that the general idea is that clients should be able to track the latest configuration and do something when it changes. For example if the server to which the client is connected goes away, it should have a way to query the system for the new list of servers and connect to one of them. The method you described is one option, outlined in the reconfig manual (right at the end there's some code for this). It would be nice to automate these steps for the user and to support a user defined policy instead of these steps (see some ideas in ZOOKEEPER-2016). For example, if you want the user to only connect to near-by servers, you'd need to filter the new list of servers before calling updateServerList. This can be done as part of a user-supplied policy.
17. First stab on implementing basics outlined in the design. Patch is not ready (pending docs update, more tests, and C client work), upload here for preview and feedback only.
18. **body:** Fixed C client unit tests, added more documentation regarding API changes, new configuration option, and ACLs around config znode. Added more Java tests. Patch is ready for review.
    label: documentation
19. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12826240/ZOOKEEPER-2014.patch against trunk revision 1757584. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 37 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3376//console This message is automatically generated.
20. build bot complains, cancel to investigate..
21. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12826264/ZOOKEEPER-2014.patch against trunk revision 1757584. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 37 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-

Build/3377//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3377//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3377//console This message is automatically generated.
22. For reviewers: https://reviews.apache.org/r/51546/
23. **body:** Fix find bug warnings.
    label: code-design
24. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12826655/ZOOKEEPER-2014.patch against trunk revision 1757584. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 37 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3378//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3378//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3378//console This message is automatically generated.
25. Nudge nudge - anyone has cycles to take a look at the patch and provide feedback? cc [~shralex], [~rgs], [~fpj], [~phunt]
26. I'll get to it before the end of this week, this is really important for the progress of the 3.5 branch.
27. Update patch to address review comments. Major changes comparing to previous version patch: * Warn user when skipACL is set and reconfig is performed. * New keeper exception that represents the error user got when doing reconfig but with reconfig feature disabled. * ZooKeeperAdmin moved to admin package. * Doc updates. * Misc updates (rename method in java way, remove tabs, etc.).
28. Thanks a lot [~fpj] for your review comments. Just upload a new patch that address all of your review comments (minors the C client ones). The remaining work is on C client side where I plan to add more tests as Java side, which will be done in next few days and then I will upload final patch for review.
29. Thanks [~hanm] for working on this. I've added few comments in RB, please take a look at it.
30. Fixed a couple of things pointed out by Rakesh (C client work is still WIP, not included here.).
31. I'll wait until you have the C client changes to make another pass, unless you need feedback on some specific parts to make progress. [~hanm]
32. Update patch to include C client work: * Removed skipACL, set up servers with super user digest auth like Java tests did. * Updated C client tests, add one specific test case to cover various reconfig failure cases caused by failure of auth. * Misc update for first review feedback related to method signature and so on.
33. [~fpj]: Sounds good - actually just updated patch that incorporates some C side changes. PTAL when you have a chance, thanks a lot!
34. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12830808/ZOOKEEPER-2014.patch against trunk revision ec20c5434cc8a334b3fd25e27d26dccf4793c8f3. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 40 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3458//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3458//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3458//console This message is automatically generated.
35. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12830832/ZOOKEEPER-2014.patch against trunk revision ec20c5434cc8a334b3fd25e27d26dccf4793c8f3. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 40 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3459//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3459//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3459//console This message is automatically generated.
36. Update patch that fixes a couple of places pointed by Flavio in review board.
37. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12831433/ZOOKEEPER-2014.patch against trunk revision ec20c5434cc8a334b3fd25e27d26dccf4793c8f3. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 42 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3462//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3462//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3462//console This message is automatically generated.
38. [~hanm] One of the tests that failed in the QA run is modified in the patch: {{testQuorumSystemChange}}. Could you have a look, please?
39. **body:** [~fpj]: This is a flaky test, and it happened before this patch with exact same error message in Apache builds and precommit build. I did a search over my email archive and found a couple of instances (e.g. ZooKeeper_branch35_openjdk7 - Build # 159). I'll create a separate JIRA and put it under ZOOKEEPER-2135. On a side note, I have an internal Jenkins job that set up and stress test

my patch branch of ZOOKEEPER-2014. There is no test failure except the one that's already tracked in ZOOKEEPER-2080. So this flaky testQuorumSystemChange one might also hard to reproduce.
label: test

40. [~hanm] I'm probably missing something obvious, but when I try to reconfigure, I get an auth failure, but I think I'm setting up everything right. This is the output of ZooKeeperMain plus my commands: {noformat} Connecting to localhost:2181 log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper). log4j:WARN Please initialize the log4j system properly. log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info. Welcome to ZooKeeper! JLine support is disabled WATCHER:: WATCHER:: WatchedEvent state:SyncConnected type:None path:null WatchedEvent state:SyncConnected type:None path:null addauth digest super:test getAcl /zookeeper/config 'digest,'super:D/InIHSb7yEEbrWz8b9l71RjZJU= : a reconfig -add server.5=127.0.0.1:1234:1235;1236 Authentication is not valid : {noformat} I have {{reconfigEnabled=true}} in the configuration of the servers.
41. [~fpj] It's a bug, thanks for pointing it out! When we do reconfigure through command line, the {{addauth}} command will set the auth packet for the connection associated with the {{ZooKeeper}} object of the {{CliCommand}}. The reconfig command however is going to be executed by the {{ZooKeeperAdmin}} object of the {{CliCommand}}, and because addauth currently does not associate the auth packet to the underlying connection managed by {{ZooKeeperAdmin}}, when reconfig request hits server it will miss auth info, thus yield auth failure. The fix is to teach {{AddAuthCommand}} to set auth packet for {{ZooKeeperAdmin}} as well. As we previously discussed in review board, an alternative is to have a single ZooKeeper object (since ZooKeeperAdmin is also a ZooKeeper) maintained in {{CliCommand}} so every command / request will converge to a single path. I did not do that for stability purposes - with a separate ZooKeeperAdmin object all existing commands (except reconfig) should not be impacted at all. I might be over cautious on this one though and maintaining a single ZK object inside CliCommand might be a better solution overall. Please let me know if you prefer one over the other. Attaching updated patch (with one line change) that fixes the issue. Verified with zkCli.sh / reconfig command on a local 3 server cluster.
42. Trivial update that fixes an important bug pointed out by Flavio (reconfig failed with NoAuth via zkCli.sh).
43. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12832546/ZOOKEEPER-2014.patch against trunk revision f78061aafb19b102c37cb6d744ec6258d5f5b66e. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 42 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3480//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3480//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3480//console This message is automatically generated.
44. **body:** Thanks for fixing it. I see the point of stability, but I don't think it is much of a concern in this case. You're just building on top of the ZooKeeper, you won't be re-wiring the existing ZooKeeper client code, I think. If I'm right, then it is better to have a single connection to the service rather than two, one for the ZooKeeper object and another for ZooKeeperAdmin. In any case, it might be worth giving it a shot to get a sense of whether it'd cause trouble.
label: code-design
45. I checked that it works for me when reconfig is enabled. I have a couple of other things I wanted to raise: # When I tried with reconfig disabled, I got this message: {noformat} reconfig -add server.5=127.0.0.1:1234:1235;1236 KeeperErrorCode = Reconfig is disabled for {noformat} And it should be only {{Reconfig disabled}}, unless we want to convey some other information. # I have also verified that to get the reconfig command to go through we only need the leader to have {{reconfigEnabled = true}}. There is no way around it unless the replicas coordinate to use the same value. We need it well documented, though.
46. Yes keeping a single connection is right thing to do. Patch on the way.
47. **body:** Patch updates: * Consolidate ZooKeeper and ZooKeeperAdmin object usage in CliCommand stack. * Fix confusing KeeperException error message when path is set as empty. * More documentation on having a consistent setting for the reconfigEnabled option across ensemble.
label: documentation
48. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12832770/ZOOKEEPER-2014.patch against trunk revision f78061aafb19b102c37cb6d744ec6258d5f5b66e. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 45 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 2 new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3481//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3481//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3481//console This message is automatically generated.
49. **body:** Fix findbug warnings.
label: code-design
50. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12832782/ZOOKEEPER-2014.patch against trunk revision f78061aafb19b102c37cb6d744ec6258d5f5b66e. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 45 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3483//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3483//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3483//console This message is automatically generated.

51. **body:** Failed tests are known flaky..
    label: test
52. Thanks for the updates, [~hanm]. I want to do a few manual checks, but otherwise, the changes look good. I encourage others to have a look as well before we check it in.
53. Upload new patch to address Abe's review comments. * Don't swallow the KeeperException when setting ACL on the config node during DataTree creation, instead let the exception bubble up to stop ZK server startup. This is to make sure we always have access controlled config node after ZK server is started. * The rest of changes are really mechanical changes due to the exception specification changes in signature. * Passed internal stress test of all unit tests.
54. +1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12833880/ZOOKEEPER-2014.patch against trunk revision cef5978969bedfe066f903834a9ea4af6d508844. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 96 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3493//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3493//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3493//console This message is automatically generated.
55. GitHub user hanm opened a pull request: https://github.com/apache/zookeeper/pull/94 ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. This PR implements ZOOKEEPER-2014. For details, please refer to * JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 * Previous review board: https://reviews.apache.org/r/51546/ You can merge this pull request into a Git repository by running: $ git pull https://github.com/hanm/zookeeper ZOOKEEPER-2014 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/zookeeper/pull/94.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #94 ---- commit 616e1275ac38890c2bf1e3ac27465172cf1c52d5 Author: Michael Han <hanm@cloudera.com> Date: 2016-10-27T16:16:27Z ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. ----
56. Github user fpj commented on the issue: https://github.com/apache/zookeeper/pull/94 @hanm since there is no comment, would you mind closing this PR and resubmitting it to see if QA picks it up?
57. Thank you [~hanm] for the continuous effort. Sorry for the delay in reviews. I've looked at the latest patch and added few minor comments in the RB. +1 from me after addressing these comments.
58. Github user hanm commented on the issue: https://github.com/apache/zookeeper/pull/94 @fpj Sure, closing and resubmitting with a new one that also addressing Rakesh's comments today.
59. Github user hanm closed the pull request at: https://github.com/apache/zookeeper/pull/94
60. GitHub user hanm opened a pull request: https://github.com/apache/zookeeper/pull/96 ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. This PR implements ZOOKEEPER-2014. For details, please refer to JIRA: https://issues.apache.org/jira/browse/ZOOKEEPER-2014 Review board: https://reviews.apache.org/r/51546/ You can merge this pull request into a Git repository by running: $ git pull https://github.com/hanm/zookeeper ZOOKEEPER-2014 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/zookeeper/pull/96.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #96 ---- commit 6d18cffde99d4cf5298e045d6c0f23b36fd62925 Author: Michael Han <hanm@cloudera.com> Date: 2016-10-31T03:58:11Z ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. ----
61. Thank you [~rakeshr] for your code review. Updated patch and PR to address your review comments. Will create a new JIRA to track the discrepancy between C / Java client regarding error code mismatch.
62. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12836113/ZOOKEEPER-2014.patch against trunk revision f6349d16fcd5f04b560095417fd2a1813ac3e855. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 96 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3504//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3504//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3504//console This message is automatically generated.
63. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12836113/ZOOKEEPER-2014.patch against trunk revision f6349d16fcd5f04b560095417fd2a1813ac3e855. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 96 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3507//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3507//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3507//console This message is automatically generated.
64. **body:** just a curious observer: why are we propagating the NoNodeException everywhere? i wasn't clear from the patch why that suddenly popped up as part of the change.
    label: code-design
65. NoNodeException is thrown from within DataTree's constructor, [addConfigNode|https://github.com/apache/zookeeper/pull/96/files#diff-a676d93082759105dd8c79c0a76a8007R264]. The intention is in cases such exception happens, we bubble exception up to abort the server startup instead of letting the server starting up with

incorrect ACLs configured on the config node. It's unfortunate that the exception has to be thrown from DataTree's constructor which is on the critical path that leads to NoNodeException being scattered everywhere.

66. ah that makes sense. i didn't dig deep enough :) it is sad that an exception "that should never happen" has such a big impact on the code. shouldn't we have thrown a runtime exception? i think it would have eliminated a lot of this patch... this is just an observation not a vote :)
67. **body:** Good point on throwing an unchecked exception which does not contaminate method signatures. The benefit of throwing a KeeperException here is minimum as the higher level code in ZooKeeperServerMain that processed typed exceptions currently does not specifically react to KeeperException (and there seems not much need to do so), so processing a KeeperException in ZooKeeperServerMain will end up with same code path as a RuntimeException. Let me update the patch again to keep it lean.
label: code-design
68. **body:** Address Ben's comment by replacing KeeperException.NoNodeException with RuntimeException in addConfigNode to avoid changing function signatures across code base.
label: code-design
69. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12837178/ZOOKEEPER-2014.patch against trunk revision bcb07a09b06c91243ed244f04a71b8daf629e286. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 62 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 20 new Findbugs (version 3.0.1) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3516//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3516//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3516//console This message is automatically generated.
70. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12837180/ZOOKEEPER-2014.patch against trunk revision bcb07a09b06c91243ed244f04a71b8daf629e286. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 60 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 20 new Findbugs (version 3.0.1) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3517//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3517//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3517//console This message is automatically generated.
71. Github user rgs1 commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86673398 --- Diff: src/c/tests/TestReconfigServer.cc --- @@ -324,4 +336,109 @@ testRemoveConnectedFollower() { zookeeper_close(zk); } +/** + * ZOOKEEPER-2014: only admin or users who are explicitly granted permission can do reconfig. + */ +void TestReconfigServer:: +testReconfigFailureWithoutAuth() { + std::vector<std::string> servers; + std::string version; + struct Stat stat; + int len = 1024; + char buf[len]; + + // connect to a follower. + int32_t leader = getLeader(); + std::vector<int32_t> followers = getFollowers(); + CPPUNIT_ASSERT(leader >= 0); + CPPUNIT_ASSERT_EQUAL(NUM_SERVERS - 1, (uint32_t)(followers.size())); + std::stringstream ss; + for (int i = 0; i < followers.size(); i++) { + ss << cluster_[followers[i]]->getHostPort() << ","; + } + ss << cluster_[leader]->getHostPort(); + std::string hosts = ss.str().c_str(); + zoo_deterministic_conn_order(true); + zhandle_t* zk = zookeeper_init(hosts.c_str(), NULL, 10000, NULL, NULL, 0); + CPPUNIT_ASSERT_EQUAL(true, waitForConnected(zk, 10)); + + std::string connectedHost(zoo_get_current_server(zk)); + std::string portString = connectedHost.substr(connectedHost.find(":") + 1); + uint32_t port; + std::istringstream (portString) >> port; + CPPUNIT_ASSERT_EQUAL(cluster_[followers[0]]->getClientPort(), port); + + // remove the follower. + len = 1024; + ss.str(""); + ss << followers[0]; + // No auth, should fail. + CPPUNIT_ASSERT_EQUAL((int)ZNOAUTH, zoo_reconfig(zk, NULL, ss.str().c_str(), NULL, -1, buf, &len, &stat)); + // Wrong auth, should fail. + CPPUNIT_ASSERT_EQUAL((int)ZOK, zoo_add_auth(zk, "digest", "super:wrong", 11, NULL,(void*)ZOK)); + CPPUNIT_ASSERT_EQUAL((int)ZNOAUTH, zoo_reconfig(zk, NULL, ss.str().c_str(), NULL, -1, buf, &len, &stat)); + // Right auth, should pass. + CPPUNIT_ASSERT_EQUAL((int)ZOK, zoo_add_auth(zk, "digest", "super:test", 10, NULL,(void*)ZOK)); + CPPUNIT_ASSERT_EQUAL((int)ZOK, zoo_reconfig(zk, NULL, ss.str().c_str(), NULL, -1, buf, &len, &stat)); + CPPUNIT_ASSERT_EQUAL((int)ZOK, zoo_getconfig(zk, 0, buf, &len, &stat)); + parseConfig(buf, len, servers, version); + CPPUNIT_ASSERT_EQUAL(NUM_SERVERS - 1, (uint32_t)(servers.size())); + for (int i = 0; i < cluster_.size(); i++) { + if (i == followers[0]) { + continue; + } + CPPUNIT_ASSERT(std::find(servers.begin(), servers.end(), + cluster_[i]->getServerString()) != servers.end()); + } + zookeeper_close(zk); +} + +void TestReconfigServer:: +testReconfigFailureWithoutServerSuperuserPasswordConfigured() { + std::vector<std::string> servers; + std::string version; + struct Stat stat; + int len = 1024; + char buf[len]; + + // Create a new quorum with the super user's password not configured. + tearDown(); + ZooKeeperQuorumServer::tConfigPairs configs; + configs.push_back(std::make_pair("reconfigEnabled", "true")); + cluster_ = ZooKeeperQuorumServer::getCluster(NUM_SERVERS, configs, ""); + + // connect to a follower. + int32_t leader = getLeader(); + std::vector<int32_t> followers = getFollowers(); + CPPUNIT_ASSERT(leader >= 0); + CPPUNIT_ASSERT_EQUAL(NUM_SERVERS - 1, (uint32_t)(followers.size())); + std::stringstream ss; + for (int i = 0; i < followers.size(); i++) { + ss << cluster_[followers[i]]->getHostPort() << ","; + } + ss << cluster_[leader]->getHostPort(); + std::string hosts = ss.str().c_str(); + zoo_deterministic_conn_order(true); + zhandle_t* zk = zookeeper_init(hosts.c_str(), NULL, 10000, NULL, NULL, 0); + CPPUNIT_ASSERT_EQUAL(true, waitForConnected(zk, 10)); + + std::string connectedHost(zoo_get_current_server(zk)); + std::string portString = connectedHost.substr(connectedHost.find(":") + 1); + uint32_t port; + std::istringstream (portString) >> port; + CPPUNIT_ASSERT_EQUAL(cluster_[followers[0]]->getClientPort(), port); --- End diff -- connect to follower seems repeated, can we move it to a private helper method?
72. Github user rgs1 commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86673475 --- Diff: src/java/main/org/apache/zookeeper/server/DataTree.java --- @@ -245,15 +245,25 @@ public DataTree() { addConfigNode(); } - public void addConfigNode() { - DataNode zookeeperZnode = nodes.get(procZookeeper); - if (zookeeperZnode!=null) { // should always be the case - zookeeperZnode.addChild(configChildZookeeper); - } else { - LOG.error("There's no /zookeeper znode

- this should never happen"); - } - nodes.put(configZookeeper, configDataNode); - } + public void addConfigNode() { + DataNode zookeeperZnode = nodes.get(procZookeeper); + if (zookeeperZnode!=null) { // should always be the case --- End diff -- nit: spaces around `!=`

73. Github user rgs1 commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86673489 --- Diff: src/java/main/org/apache/zookeeper/server/DataTree.java --- @@ -245,15 +245,25 @@ public DataTree() { addConfigNode(); } - public void addConfigNode() { - DataNode zookeeperZnode = nodes.get(procZookeeper); - if (zookeeperZnode!=null) { // should always be the case - zookeeperZnode.addChild(configChildZookeeper); - } else { - LOG.error("There's no /zookeeper znode - this should never happen"); - } - nodes.put(configZookeeper, configDataNode); - } + public void addConfigNode() { + DataNode zookeeperZnode = nodes.get(procZookeeper); + if (zookeeperZnode!=null) { // should always be the case --- End diff -- i think this should be an assertion (and we can drop the LOG.error call)
74. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86679075 --- Diff: src/java/test/org/apache/zookeeper/test/ClientBase.java --- @@ -356,7 +356,7 @@ private static int getPort(String hostPort) { */ public static void startServerInstance(File dataDir, ServerCnxnFactory factory, String hostPort) throws IOException, - InterruptedException { + InterruptedException, KeeperException.NoNodeException { --- End diff -- this isn't needed anymore right?
75. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86679034 --- Diff: src/java/main/org/apache/zookeeper/server/DataTree.java --- @@ -245,15 +245,25 @@ public DataTree() { addConfigNode(); } - public void addConfigNode() { - DataNode zookeeperZnode = nodes.get(procZookeeper); - if (zookeeperZnode!=null) { // should always be the case - zookeeperZnode.addChild(configChildZookeeper); - } else { - LOG.error("There's no /zookeeper znode - this should never happen"); - } - nodes.put(configZookeeper, configDataNode); - } + public void addConfigNode() { + DataNode zookeeperZnode = nodes.get(procZookeeper); + if (zookeeperZnode!=null) { // should always be the case + zookeeperZnode.addChild(configChildZookeeper); + } else { + LOG.error("There's no /zookeeper znode - this should never happen."); + } + + nodes.put(configZookeeper, configDataNode); + try { + // Reconfig node is access controlled by default (ZOOKEEPER-2014). + setACL(configZookeeper, ZooDefs.Ids.READ_ACL_UNSAFE, -1); + } catch (KeeperException.NoNodeException e) { + LOG.error("Fail to set ACL on {} - this should never happen: {}", configZookeeper, e); --- End diff -- actually if we are asserting above, perhaps we should also assert here.
76. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86678939 --- Diff: src/docs/src/documentation/content/xdocs/zookeeperAdmin.xml --- @@ -1108,6 +1109,42 @@ server.3=zoo3:2888:3888</programlisting> </para> </listitem> </varlistentry> + + <varlistentry> + <term>reconfigEnabled</term> + + <listitem> + <para>(No Java system property)</para> + + <para><emphasis role="bold">New in 3.5.3:</emphasis> + This controls the enabling or disabling of + <ulink url="zookeeperReconfig.html"> + Dynamic Reconfiguration</ulink> feature. When the feature + is enabled, users can perform reconfigure operations through + the ZooKeeper client API or through ZooKeeper command line tools + assuming users are authorized to perform such operations. + When the feature is disabled, no user, including the super user, + can perform a reconfiguration. Any attempt to reconfigure will return an error. + <emphasis role="bold">"reconfigEnabled"</emphasis> option can be set as + <emphasis role="bold">"reconfigEnabled=false"</emphasis> or + <emphasis role="bold">"reconfigEnabled=true"</emphasis> + to a server's config file, or using QuorumPeerConfig's + setReconfigEnabled method. The default value is false. + + If present, the value should be consistent across every server in + the entire ensemble. Setting the value as true on some servers and false + on other servers will cause inconsistent behavior depends on which server --- End diff -- depends -> depending
77. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86679106 --- Diff: src/java/test/org/apache/zookeeper/test/ReconfigExceptionTest.java --- @@ -0,0 +1,220 @@ +/** + * Licensed to the Apache Software Foundation (ASF) under one + * or more contributor license agreements. See the NOTICE file + * distributed with this work for additional information + * regarding copyright ownership. The ASF licenses this file + * to you under the Apache License, Version 2.0 (the + * "License"); you may not use this file except in compliance + * with the License. You may obtain a copy of the License at + * + * http://www.apache.org/licenses/LICENSE-2.0 + * + * Unless required by applicable law or agreed to in writing, software + * distributed under the License is distributed on an "AS IS" BASIS, + * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. + * See the License for the specific language governing permissions and + * limitations under the License. + */ + +package org.apache.zookeeper.test; + +import java.io.IOException; +import java.util.ArrayList; +import java.util.Collections; +import java.util.List; +import java.util.concurrent.TimeoutException; + +import org.apache.zookeeper.ZKTestCase; +import org.apache.zookeeper.KeeperException; +import org.apache.zookeeper.ZooDefs; +import org.apache.zookeeper.PortAssignment; +import org.apache.zookeeper.admin.ZooKeeperAdmin; +import org.apache.zookeeper.data.ACL; +import org.apache.zookeeper.data.Id; +import org.apache.zookeeper.data.Stat; +import org.apache.zookeeper.server.quorum.QuorumPeerConfig; +import org.junit.After; +import org.junit.Assert; +import org.junit.Before; +import org.junit.Test; +import org.slf4j.Logger; +import org.slf4j.LoggerFactory; + +public class ReconfigExceptionTest extends ZKTestCase { + private static final Logger LOG = LoggerFactory + .getLogger(ReconfigExceptionTest.class); + private static String authProvider = + "zookeeper.DigestAuthenticationProvider.superDigest"; + // Use DigestAuthenticationProvider.base64Encode or + // run ZooKeeper jar with org.apache.zookeeper.server.auth.DigestAuthenticationProvider to generate password. + // An example: + // java -cp zookeeper-3.6.0-SNAPSHOT.jar:lib/log4j-1.2.17.jar:lib/slf4j-log4j12-1.7.5.jar: + // lib/slf4j-api-1.7.5.jar org.apache.zookeeper.server.auth.DigestAuthenticationProvider super:test + // The password here is 'test'. + private static String superDigest = "super:D/InIHSb7yEEbrWz8b9l71RjZJU="; + private QuorumUtil qu; + private ZooKeeperAdmin zkAdmin; + + @Before + public void setup() throws InterruptedException, KeeperException.NoNodeException { --- End diff -- +1 awesome work! i only found little nits! thanx for sticking with this!
78. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86679094 --- Diff: src/java/test/org/apache/zookeeper/test/ReconfigExceptionTest.java --- @@ -0,0 +1,220 @@ +/** + * Licensed to the Apache Software Foundation (ASF) under one + * or more contributor license agreements. See the NOTICE file + * distributed with this work for additional information + * regarding copyright ownership. The ASF licenses this file + * to you under the Apache License, Version 2.0 (the + * "License"); you may not use this file except in compliance + * with the License. You may obtain a copy of the License at + * + * http://www.apache.org/licenses/LICENSE-2.0 + * + * Unless required by applicable law or agreed to in writing, software + * distributed under the License is distributed on an "AS IS" BASIS, + * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. + * See the License for the specific language governing permissions and + * limitations under the License. + */ + +package org.apache.zookeeper.test; + +import java.io.IOException; +import java.util.ArrayList; +import java.util.Collections; +import java.util.List; +import java.util.concurrent.TimeoutException; + +import org.apache.zookeeper.ZKTestCase; +import org.apache.zookeeper.KeeperException; +import org.apache.zookeeper.ZooDefs;

+import org.apache.zookeeper.PortAssignment; +import org.apache.zookeeper.admin.ZooKeeperAdmin; +import org.apache.zookeeper.data.ACL; +import org.apache.zookeeper.data.Id; +import org.apache.zookeeper.data.Stat; +import org.apache.zookeeper.server.quorum.QuorumPeerConfig; +import org.junit.After; +import org.junit.Assert; +import org.junit.Before; +import org.junit.Test; +import org.slf4j.Logger; +import org.slf4j.LoggerFactory; + +public class ReconfigExceptionTest extends ZKTestCase { + private static final Logger LOG = LoggerFactory + .getLogger(ReconfigExceptionTest.class); + private static String authProvider = "zookeeper.DigestAuthenticationProvider.superDigest"; + // Use DigestAuthenticationProvider.base64Encode or + // run ZooKeeper jar with org.apache.zookeeper.server.auth.DigestAuthenticationProvider to generate password. + // An example: + // java -cp zookeeper-3.6.0-SNAPSHOT.jar:lib/log4j-1.2.17.jar:lib/slf4j-log4j12-1.7.5.jar: + // lib/slf4j-api-1.7.5.jar org.apache.zookeeper.server.auth.DigestAuthenticationProvider super:test + // The password here is 'test'. + private static String superDigest = "super:D/InIHSb7yEEbrWz8b9l71RjZJU="; + private QuorumUtil qu; + private ZooKeeperAdmin zkAdmin; + + @Before + public void setup() throws InterruptedException, KeeperException.NoNodeException { --- End diff -- we don't throw NoNodeException anymore

79. Github user breed commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r86679058 --- Diff: src/java/test/org/apache/zookeeper/server/DataTreeTest.java --- @@ -200,29 +198,34 @@ public void testSerializeDoesntLockDataNodeWhileWriting() throws Exception { BinaryOutputArchive oa = new BinaryOutputArchive(out) { @Override public void writeRecord(Record r, String tag) throws IOException { - DataNode node = (DataNode) r; - if (node.data.length == 1 && node.data[0] == 42) { - final Semaphore semaphore = new Semaphore(0); - new Thread(new Runnable() { - @Override - public void run() { - synchronized (markerNode) { - //When we lock markerNode, allow writeRecord to continue - semaphore.release(); + // Need check if the record is a DataNode instance because of changes in ZOOKEEPER-2014 + // which adds default ACL to config node. + if (r instanceof DataNode) { --- End diff -- is there a reason we added the instanceof here? if we didn't need it before, why do we need it now?
80. Update patch for addressing review comments from Ben and Raul.
81. Github user hanm commented on the issue: https://github.com/apache/zookeeper/pull/96 Thanks @breed and @rgs1 for your time and review feedback. Pull request, patch, and review board is now updated.
82. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12837643/ZOOKEEPER-2014.patch against trunk revision bcb07a09b06c91243ed244f04a71b8daf629e286. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 57 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 20 new Findbugs (version 3.0.1) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3524//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3524//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/3524//console This message is automatically generated.
83. shall i commit it or are we waiting on something else?
84. [~breed] We were previously waiting for ZOOKEEPER-2624 to land and for more review comments. Now ZOOKEEPER-2624 landed and more review comments were collected and addressed, I think this one is ready to land :-). cc [~fpj].
85. Github user fpj commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r87536973 --- Diff: src/java/test/org/apache/zookeeper/server/DataTreeTest.java --- @@ -200,29 +198,34 @@ public void testSerializeDoesntLockDataNodeWhileWriting() throws Exception { BinaryOutputArchive oa = new BinaryOutputArchive(out) { @Override public void writeRecord(Record r, String tag) throws IOException { - DataNode node = (DataNode) r; - if (node.data.length == 1 && node.data[0] == 42) { - final Semaphore semaphore = new Semaphore(0); - new Thread(new Runnable() { - @Override - public void run() { - synchronized (markerNode) { - //When we lock markerNode, allow writeRecord to continue - semaphore.release(); + // Need check if the record is a DataNode instance because of changes in ZOOKEEPER-2014 + // which adds default ACL to config node. + if (r instanceof DataNode) { --- End diff -- @hanm hmm, I'm not sure about this. In the changes for `DataTree`, we only set the ACL of the `/zookeeper/config` znode, but setting ACLs was something we were doing before, so I'm confused about why we can have a mix of znode records and ACL records with the changes proposed here. Could you clarify, please?
86. Github user hanm commented on a diff in the pull request: https://github.com/apache/zookeeper/pull/96#discussion_r87543684 --- Diff: src/java/test/org/apache/zookeeper/server/DataTreeTest.java --- @@ -200,29 +198,34 @@ public void testSerializeDoesntLockDataNodeWhileWriting() throws Exception { BinaryOutputArchive oa = new BinaryOutputArchive(out) { @Override public void writeRecord(Record r, String tag) throws IOException { - DataNode node = (DataNode) r; - if (node.data.length == 1 && node.data[0] == 42) { - final Semaphore semaphore = new Semaphore(0); - new Thread(new Runnable() { - @Override - public void run() { - synchronized (markerNode) { - //When we lock markerNode, allow writeRecord to continue - semaphore.release(); + // Need check if the record is a DataNode instance because of changes in ZOOKEEPER-2014 + // which adds default ACL to config node. + if (r instanceof DataNode) { --- End diff -- @fpj - we were not setting ACLs on intrinsic znodes (i.e. /zookeeper/config) ZooKeeper implicitly created while initializing a DataTree before. And for this test case, it only creates znodes, not ACLs. As a result, it's reasonable for the previous test case to assume every record that's serializing is a DataNode record. Now with this patch, there is an ACL implicitly created when /zookeeper/config node is created, so the previous assumption (that all records to be serialized are DataNode record) does not hold. Thus, a change is required. For reference, you could put a break point on https://github.com/apache/zookeeper/blob/master/src/java/main/org/apache/zookeeper/server/ReferenceCountedACLCache.java#L133 while running this test case, and you will see there is one ACL that's serialized. Now you can remove the ACL associated with /zookeeper/config at https://github.com/apache/zookeeper/pull/96/files#diff-a676d93082759105dd8c79c0a76a8007R259, and you will see the break point on ReferenceCountedACLCache.java previous set not get hit. That is the difference. Another way to experiment this is to create an ACL in this test (without applying this pull request first), something like: `final DataNode markerNode = tree.getNode("/marker"); tree.setACL("/marker", ZooDefs.Ids.READ_ACL_UNSAFE, -1);` will do. Then we will see the same type casting failure - this simulates what this PR will do in terms of changing the type of records. Basically I think the root cause is the test itself could be made more robust, by eliminate the assumptions (that every record is a DataNode) that might not always hold.
87. Github user asfgit closed the pull request at: https://github.com/apache/zookeeper/pull/96

88. +1, great work [~hanm], thanks for the patch!
89. FAILURE: Integrated in Jenkins build ZooKeeper-trunk #3155 (See [https://builds.apache.org/job/ZooKeeper-trunk/3155/]) ZOOKEEPER-2014: Only admin should be allowed to reconfig a cluster. (fpj: rev 73e102a58d01b27bc6208bbfbde2d12f0deba1f4) * (edit) src/java/main/org/apache/zookeeper/server/PrepRequestProcessor.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/ReconfigLegacyTest.java * (edit) src/java/main/org/apache/zookeeper/ZooKeeper.java * (edit) src/java/main/org/apache/zookeeper/cli/CliCommand.java * (edit) src/java/test/org/apache/zookeeper/TestableZooKeeper.java * (edit) src/java/systest/org/apache/zookeeper/test/system/BaseSysTest.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/ReconfigDuringLeaderSyncTest.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/ReconfigBackupTest.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/LearnerTest.java * (edit) src/java/main/org/apache/zookeeper/KeeperException.java * (edit) src/java/main/org/apache/zookeeper/server/quorum/QuorumPeerMain.java * (edit) src/java/test/org/apache/zookeeper/server/DataTreeTest.java * (edit) src/c/tests/ZooKeeperQuorumServer.h * (add) src/java/test/org/apache/zookeeper/test/ReconfigMisconfigTest.java * (edit) src/docs/src/documentation/content/xdocs/zookeeperAdmin.xml * (edit) src/java/main/org/apache/zookeeper/cli/ReconfigCommand.java * (edit) src/c/tests/TestReconfigServer.cc * (add) src/java/test/org/apache/zookeeper/test/ReconfigExceptionTest.java * (edit) src/c/include/zookeeper.h * (edit) src/java/test/org/apache/zookeeper/server/quorum/RaceConditionTest.java * (add) src/java/main/org/apache/zookeeper/admin/ZooKeeperAdmin.java * (edit) src/java/test/org/apache/zookeeper/test/ACLTest.java * (edit) build.xml * (edit) src/java/main/org/apache/zookeeper/ZooKeeperMain.java * (edit) src/java/main/org/apache/zookeeper/server/DataTree.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/ReconfigFailureCasesTest.java * (edit) src/java/main/org/apache/zookeeper/server/ZooKeeperServer.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/Zab1_0Test.java * (edit) src/java/main/org/apache/zookeeper/server/ZooKeeperServerMain.java * (edit) src/java/test/org/apache/zookeeper/test/ReconfigTest.java * (edit) src/docs/src/documentation/content/xdocs/zookeeperReconfig.xml * (edit) src/java/main/org/apache/zookeeper/server/quorum/QuorumPeerConfig.java * (edit) src/java/test/org/apache/zookeeper/server/quorum/StandaloneDisabledTest.java * (edit) src/java/test/org/apache/zookeeper/test/StandaloneTest.java * (edit) src/c/tests/ZooKeeperQuorumServer.cc * (edit) src/java/main/org/apache/zookeeper/ClientCnxn.java
90. **body:** I realize I'm very late to this issue but I truly don't understand the benefit of this. This change has completely broken Curator and I'm now struggling to figure out how to fix it. How does breaking all existing clients help ZooKeeper usage?
    label: code-design
91. I'm terribly sorry I was so late to this issue. Now that it's released I see even more problems. I just sent this email to @dev {panel} reconfig() is limited to "super" user. Perversely, this reduces security as "super" user is utterly insecure. Requiring new databases to be post-applied via super user creates a security hole. For the time that the new ACLs for /zookeeper/config are to be changed the ZooKeeper instance will be in "super" user mode. Additionally, having to do all this is terribly cumbersome. Lastly, the docs only make passing mention of this. I think users will be very surprised by this - especially as the docs refer users to ReconfigExceptionTest.java which isn't part of the client distribution. {panel}