[Tutorial] Apache ZooKeeper – Setting ACL in ZooKeeper Client

Posted on July 24, 2014 by ihong5 • Tagged ZooKeeper, zookeeper acl • 6 Comments

Now let's talk about setting the ACL of a znode in ZooKeeper. Before getting into the details, let's talk more about the scheme and ID.

1. Scheme and ID

ID, as name suggests, is an identifier comprised of a username and password. By default, when the znode has an ACL set accessible by a specific group of users or an individual, the `<username>:<password>` is first hashed using an SHA-1 hashing algorithm, and then it (hex-string) is base-64 encoded.

As mentioned in earlier blog entries, scheme is like a group of users that are authorized to access a certain znode with a scheme-and-id-specific ACL set.

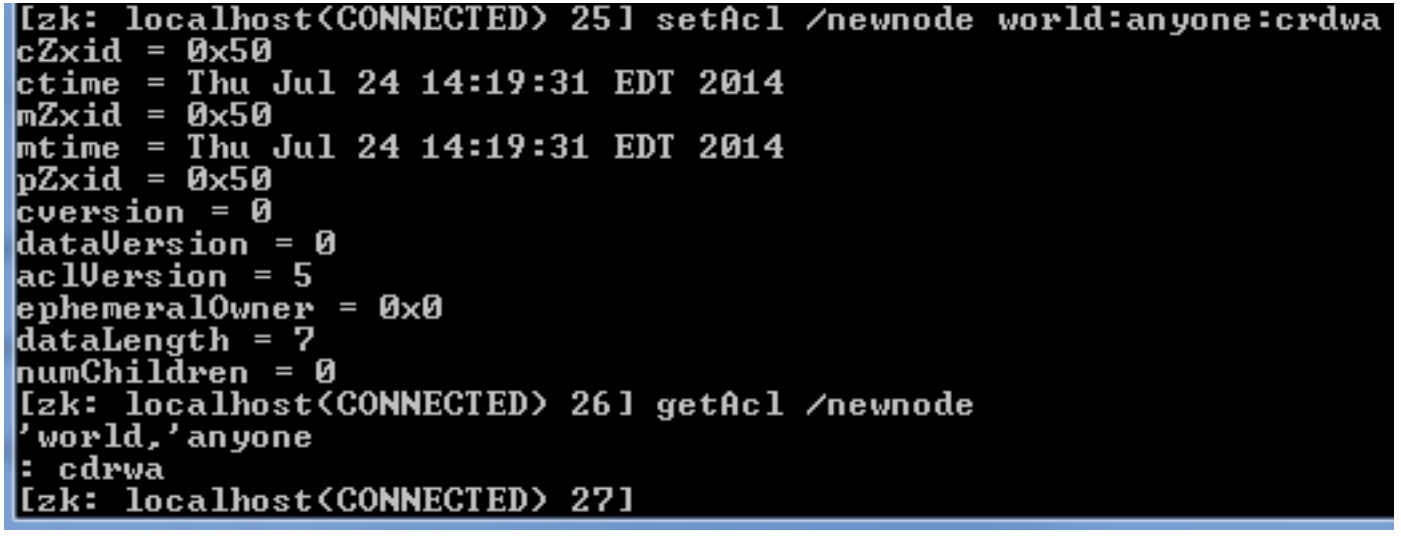
1.1. World Scheme

`World` scheme has one ID (`anyone`). This represents any user in the world.

For example, we type in the following command to set the znode accessible by anyone.

```
setAcl /newznode world:anyone:crdwa
```

By doing it correctly, You should get something like this in return:



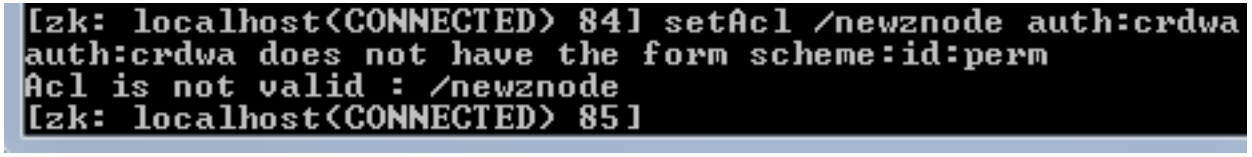
1.2. Auth Scheme

`Auth` scheme represents a *"manually"* set group of authenticated users. According to the ZooKeeper documentation

(http://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html), `auth` does not utilize any ID. Unless I am mistaken, this seems not to be the case. Because if you try to set ACL on a znode using `auth` scheme and not provide any ID, it tells you that is not a valid ID, or some form of ID is needed. Below is a **(bad)** example:

```
setAcl /newznode auth:crdwa
```

as seen above, I did not provide any form of ID. This is what I get:



A correct way to use this scheme would be as follows:

```
setAcl /newznode auth:username:password:crdwa
```

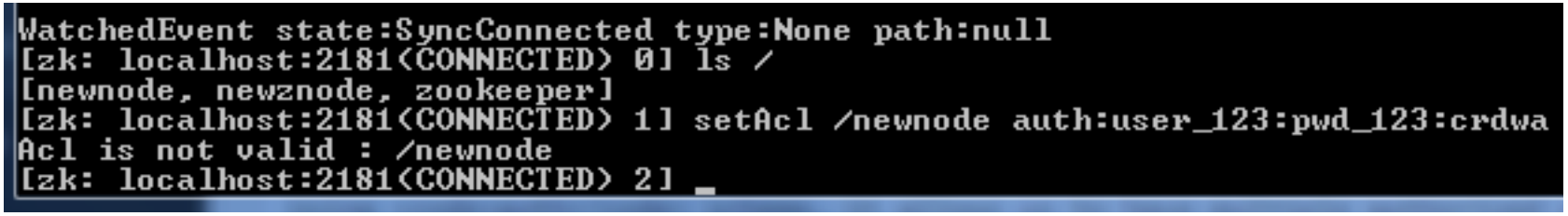
Using `auth` scheme allows us to have multiple authorized users to access a single znode with the different username and password combination. Say we have 3 users:

```
username : password
user_123 : pwd_123
user_456 : pwd_456
user_789 : pwd_789
```

We can use the same syntax above by replacing `username` with `user_123, user_456, or user_789` and `password` with `pwd_123, pwd_456, or pwd_789` respectively.

1.2.1 addauth Command

One important thing to note is you must use the `addauth` command **before** proceeding to set the ACL of a znode using the `auth` scheme. If you try to set the ACL **before** executing the `addauth` command, you will get an error as below:



Correct way to do is to execute `addauth` command first, and then execute the `setAcl` command. Below is the syntax of command execution for `addauth`:

```
addauth /<node-name> digest <username>:<password>
```

By adding the authenticator and setting ACL accordingly, you can ensure that you set the ACL correctly.

```
[zk: localhost(CONNECTED) 6] addauth digest user_123:pwd_123
[zk: localhost(CONNECTED) 7] setAcl /newnode auth:user_123:pwd_123:crdwa
cZxid = 0x50
ctime = Thu Jul 24 14:19:31 EDT 2014
mZxid = 0x50
mtime = Thu Jul 24 14:19:31 EDT 2014
pZxid = 0x50
cversion = 0
dataVersion = 0
aclVersion = 1
ephemeralOwner = 0x0
dataLength = 7
numChildren = 0
[zk: localhost(CONNECTED) 8] getAcl /newnode
'digest,'user_123:mh/MD3lrRTClTHcHDpfSgpdYoPE=
: cdrwa
[zk: localhost(CONNECTED) 9]
```

Repeat the steps for additional username and password combo, and the ACL for that `newznode` looks like this:

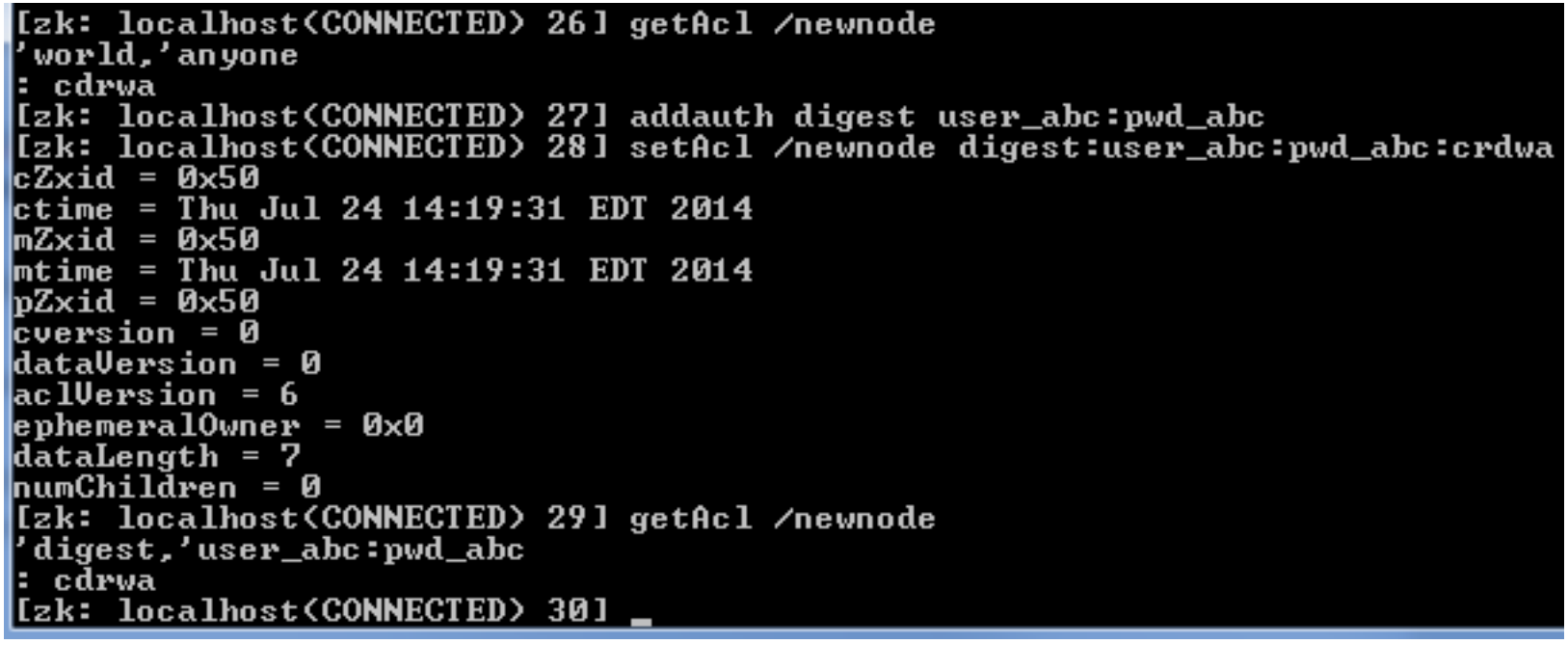
```
[zk: localhost(CONNECTED) 13] getAcl /newnode
'digest,'user_123:mh/MD3lrRTClTHcHDpfSgpdYoPE=
: cdrwa
'digest,'user_456:EINEyxG7hMOouxFxeP7U/B3drgo=
: cdrwa
'digest,'user_789:JKb5fGeoKYiNH8qQ/976Rfwni9U=
: cdrwa
[zk: localhost(CONNECTED) 14] _
```

1.3. Digest Scheme

`Digest` scheme represents an individual user with authentication. This uses `username:password` string that is hashed using the SHA-1 hashing algorithm, and that hashed string is in turn base64 encoded. According to the ZooKeeper website, it is stated that the MD5 hash of `<username>:<password>` is used as an ACL ID identity. Unless I am mistaken, that seems not to be the case. Instead, what I found was that `<username>:<Base64 encoded SHA-1`

`hash of username:password>` is used as an ACL ID (Please see above pictures under the Auth section).

What's really funny is that if I authenticate an individual user on a znode using `digest` scheme on ZooKeeper client, instead of storing the username and encoded hash string of `<username:password>` like it should, it stores a clear, human-readable text of `<username:password>` as an ID. Executing the `addauth` command before setting the ACL with `digest` scheme does not work either. Below is the picture that illustrates my point:



Unless it is easy to work backwards – decoding the `user_abc:pwd_abc`, and then take that decoded string and undo the SHA-1 hashing part, it turns out setting ACL using `digest` scheme on a znode in ZooKeeper client is pointless.

Good thing is that if you `setAcl` a znode using `digest` scheme via client, you can delete it.

1.4. Host Scheme

`Host` scheme represents anyone within the same hosting server. I have not done enough with the `host` scheme yet, but I will come back to this with more details.

1.5. IP Scheme

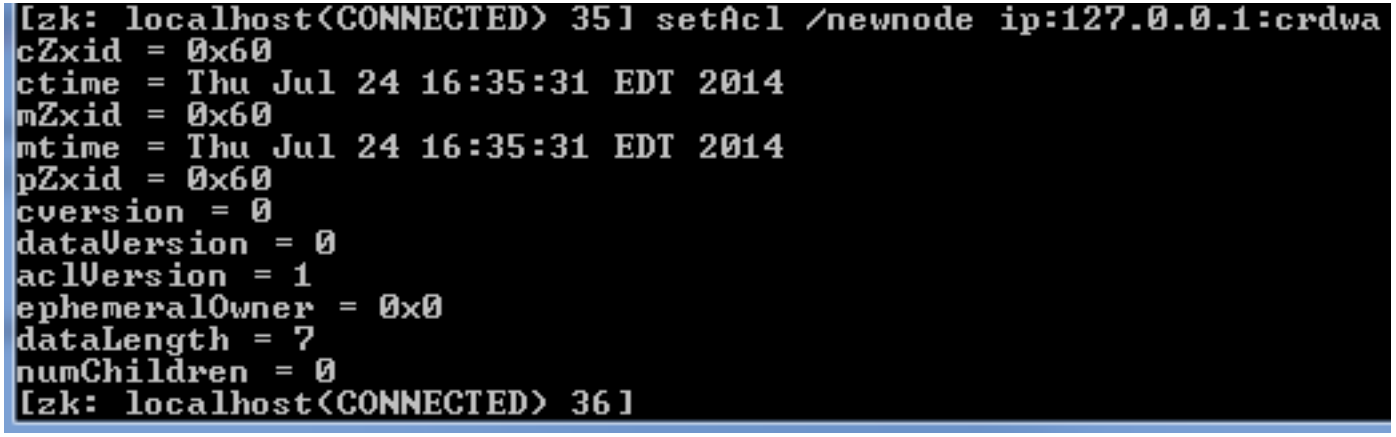
`IP` scheme represents any user within the same IP address. Easiest example to use in this case would be `127.0.0.1`, which represents the user of that any local machine, since any local machine will have `127.0.0.1` point to the `localhost`. Below is the syntax of `setAcl` using `IP` scheme:

```
setAcl /<node-name> ip:<IPv4-address>:<permission-set>
```

Using the syntax above, below is an example using the `127.0.0.1` IP address:

```
setAcl /newnode ip:127.0.0.1:crdwa
```

If done correctly, you should get the znode stat like the picture below:



That is it for now. On my next blog post, I will briefly talk about how to access them in Java; furthermore, I will talk more in detail about how username and password are stored. Thanks for reading as usual, and happy zookeeping!

Share this:

Twitter Facebook 1 Google

Loading...

6 THOUGHTS ON "[TUTORIAL] APACHE ZOOKEEPER – SETTING ACL IN ZOOKEEPER CLIENT"

sandeep says:

January 2, 2015 at 6:22 am

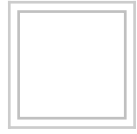
Hi I have Znode which is using ipAuthScheme , so on node multiple ips are having access now i want to remove some of ip , How i can do that ?

Mahalia says:
January 3, 2015 at 7:04 am

You post interesting articles here. Your page deserves much bigger audience.
It can go viral if you give it initial boost, i know very
useful service that can help you, simply type in google: svetsern traffic tips

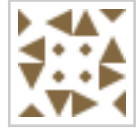
---

thefourtheye (@dFourthi) says:
November 12, 2015 at 5:45 am



Thanks a lot The official documentation is also not helping much. This post is the best I found in the internet.

Jeffrey Brill says:
January 7, 2016 at 6:07 pm



I realize this is ancient, but I found this in a google search and was briefly led astray, so I wanted to be a good citizen and clear up an

error here!

> What's really funny is that if I authenticate an individual user on a znode using digest scheme on ZooKeeper client, instead of storing the username and encoded hash string of like it should, it stores a clear, human-readable text of as an ID.

When you auth, you want to send "digest", "username:password". When you set an ACL, you're responsible for running the sha1 operation yourself- In fact, in your example, if instead of getAcl /newNode you attempted to do get /newNode, it would fail!

Instead, you would need to run an operation like this:
{hash} = b64encode(sha1(user_abc:pwd_abc))
setAcl /newnode digest:user_abc:{hash}:crdwa

As a result, getAcl will not show the password in cleartext, but a user who authenticates as user_abc:pwd_abc will be able to operate against the node.

The kazoo source is a good reference for how this is supposed to work. Enjoy!

Hu Zhanchi says:
March 11, 2016 at 1:13 am

well done

Sebastian says:

April 18, 2016 at 10:33 am

How can I change the scheme or auth or acl in a znode??

LEAVE A REPLY

Enter your comment here...

SEARCH MY BLOG

Search ...

RECENT POSTS

- Modified Consistent Hashing Rings in OpenStack Swift August 22, 2014
- Consistent Hashing Algorithm August 19, 2014
- [Tutorial] Apache ZooKeeper – Setting ACL in ZooKeeper Client July 24, 2014
- [Tutorial] Apache ZooKeeper ACL (Access Control List) Getting Permission Sets July 10, 2014
- [Tutorial] How to Install And Setup Apache ZooKeeper Standalone (Windows) July 7, 2014

CALENDAR

July 2014

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

« JUN AUG »

TAG LIBRARY

consistent hashing algorithm consistent hashing rings data integration IBM InfoSphere DataStage maven object storage

openstack openstack swift swift ZooKeeper zookeeper acl