

KIP-357: Add support to list ACLs per principal

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)

This page is meant as a template for writing a [KIP](#). To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *Adopted*

Discussion thread: [here](#)

JIRA: [here](#)

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

From [JIRA description](#)

"Currently the `kafka-acls` command has a `--list` option that can list per resource. In order to look at the ACLs for a particular principal the user needs to iterate through the entire list to figure out what privileges a particular principal has been granted. An option to list the ACL per principal would simplify this process."

We propose new optional argument "--principal" to list the ACLs for the specified principals.

Public Interfaces

kafka-acls.sh script with "--list" option supports optional argument "--principal" with the following specification

```
val listPrincipalsOpt = parser.accepts("principal", "List ACLs for the specified principal. principal is in
principalType:name format." +
  " Note that principalType must be supported by the Authorizer being used. Multiple principal options can
be passed.")
  .withOptionalArg()
  .describedAs("principal")
  .ofType(classOf[String])
```

Example:

```
>> sh kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2181 --list --principal User:User1 --principal User:User2
ACLs for principal `User:User1`
Current ACLs for resource `Group:PREFIXED:TEST_GROUP`:
    User:User1 has Allow permission for operations: Read from hosts: *

Current ACLs for resource `Topic:PREFIXED:TEST_TOPIC`:
    User:User1 has Allow permission for operations: Read from hosts: *
    User:User1 has Allow permission for operations: Create from hosts: *
    User:User1 has Allow permission for operations: Write from hosts: *
    User:User1 has Allow permission for operations: Describe from hosts: *

ACLs for principal `User:User2`
Current ACLs for resource `Group:PREFIXED:TEST_GROUP`:
    User:User2 has Allow permission for operations: Read from hosts: *

Current ACLs for resource `Topic:PREFIXED:TEST_TOPIC`:
    User:User2 has Allow permission for operations: Read from hosts: *
    User:User2 has Allow permission for operations: Create from hosts: *
    User:User2 has Allow permission for operations: Write from hosts: *
    User:User2 has Allow permission for operations: Describe from hosts: *
```

Proposed Changes

AclCommand with Authorizer Interface:

We have a method in Authorizer interface to get acls for a given principal.
We will use this method to fetch acls and filter the results for the requested Resources.

```
Authorizer {
def getAcls(principal: KafkaPrincipal): Map[Resource, Set[Acl]]
}
```

AclCommand with AdminClient API:

Currently AdminClient API doesn't have a API to fetch acls for a given principal.
So while using AclCommand with AdminClient API ([KIP-332: Update AclCommand to use AdminClient API](#)), we just filter the results returned from the describeAcls API.

We can add new AdminClient API/new DescribeAclsRequest if required in future.

Compatibility, Deprecation, and Migration Plan

- There won't be any change of current behavior.