

# KIP-332: Update AclCommand to use AdminClient API

- [Status](#)
- [Motivation](#)
- [Rejected Alternatives](#)

*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

Currently AclCommand (kafka-acls.sh) uses authorizer class (Default is SimpleAclAuthorizer) to manage ACLs. This means, when using out-of-box authorizer implementation (SimpleAclAuthorizer), users should have access to the ZooKeeper clusters and should have access to broker principal in case of Kerberised clusters.

AdminClient supports ACL management API (createAcls, deleteAcls, describeAcls). Users having necessary permissions (Alter permission on ClusterResource) can execute these API.

The proposal is to allow AclCommand to support AdminClient API based acl management. This will allow kafka-acls.sh script users to manage ACLs without interacting zookeeper/authorizer directly.

## Public Interfaces

The following changes are proposed to the kafka-acls.sh script:

1. Add a new option "--bootstrap-server" which is used to connect kafka cluster . **This option can not be used with existing "--authorizer" , "--authorizer-properties" options.**
2. Add a new option "--command-config" which takes a property file containing configs to be passed to admin Client. This option can be used with only "--bootstrap-server".
3. Users can use either "--bootstrap-server" or "--authorizer" options. By default "--authorizer" will be used, which defaults to SimpleAclAuthorizer.

## Proposed Changes

kafka-acls.sh script with "--bootstrap-server" option supports all the ACL management operations supported by existing "--authorizer" option.

```
// To add acls
>>bin/kafka-acls.sh --bootstrap-server localhost:9092 --add --allow-principal User:Bob --producer --topic Test-
topic
Adding ACLs for resource `ResourcePattern(resourceType=TOPIC, name=Test-topic, patternType=LITERAL)`:
User:Bob has Allow permission for operations: Describe from hosts: *
User:Bob has Allow permission for operations: Create from hosts: *
User:Bob has Allow permission for operations: Write from hosts: *
Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=Test-topic, patternType=LITERAL)`:
(principal=User:Bob, host=*, operation=DESCRIBE, permissionType=ALLOW)
(principal=User:Bob, host=*, operation=CREATE, permissionType=ALLOW)
(principal=User:Bob, host=*, operation=WRITE, permissionType=ALLOW)

// To list acls
>>bin/kafka-acls.sh --bootstrap-server localhost:9092 --list --topic Test-topic
Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=Test-topic, patternType=LITERAL)`:
(principal=User:Bob, host=*, operation=DESCRIBE, permissionType=ALLOW)
(principal=User:Bob, host=*, operation=CREATE, permissionType=ALLOW)
(principal=User:Bob, host=*, operation=WRITE, permissionType=ALLOW)

// To remove acls
>>bin/kafka-acls.sh --bootstrap-server localhost:9092 --remove --allow-principal User:Bob --producer --topic
Test-topic
Are you sure you want to remove ACLs:
User:Bob has Allow permission for operations: Describe from hosts: *
User:Bob has Allow permission for operations: Create from hosts: *
User:Bob has Allow permission for operations: Write from hosts: *
from resource filter `ResourcePattern(resourceType=TOPIC, name=Test-topic, patternType=LITERAL)`? (y/n)
```

## Compatibility, Deprecation, and Migration Plan

- There won't be any change of current behavior. Users can continue to use existing "--authorizer" option.

## Rejected Alternatives