

KIP-430 - Return Authorized Operations in Describe Responses


- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [Kafka Protocol Changes](#)
 - [Metadata Request and Response: v8](#)
 - [DescribeGroups Request and Response v3](#)
 - [AdminClient API Changes](#)
 - [Example: DescribeTopicsOptions](#)
 - [Example: ConsumerGroupDescription](#)
 - [ResourceType API Changes](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Changes dropped from the Proposal](#)
 - [Authorizer API Changes](#)
- [Rejected Alternatives](#)
 - [Add a new request to obtain authorized operations for different resources](#)

Status

Current state: *"Accepted"*

Discussion thread: [here](#)

JIRA:

 Unable to render Jira issues macro, execution error.

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

Motivation

When resources are described using `AdminClient` (e.g. `AdminClient#describeTopics`), the metadata returned for the resource includes all the metadata available in the broker to enable the client to perform various operations on the resource, but it does not include any information related to which operations the user requesting the metadata is authorized to perform.

Kafka protocol currently allows users with `Describe` access on the `Cluster` to describe ACLs, but even users with this access need to parse ACLs to determine what operations are permitted for the user on each resource. This requires complex logic since resources may be wildcarded or prefixed and operations or hosts may also be wildcarded.

Users with `Describe` access can determine which operations are authorized by invoking the operation and processing any resulting `AuthorizationException`. Since this information is accessible to users with `Describe` access any way, it will be good to have a simpler way for users to determine what operations they are authorized to perform on a resource that they describe. Users without `Describe` access will continue to receive errors that don't leak any information about the existence of the resource.

Public Interfaces

This KIP proposes to optionally return authorized operations in `DescribeXxx` responses of requests to describe resources managed by the broker. A new boolean field will be added to request this information. By default, this information will not be requested or returned.

Kafka Protocol Changes

The following requests and responses will be updated to request/return authorized operations:

- `Metadata` (for `Cluster` and `Topic`)
- `DescribeGroups` (for `Consumer Groups`)

The version of the relevant requests will be bumped up. Authorized operations will be returned as an `INT32` with bits set for each permitted operation. The bitfield corresponding to each operation will be the existing operation code used in `AclOperation`. If in future we exceed 32 operations and need more bits, the field can be made into `INT64` with a version bump. The highest code currently in use is 12. Broker will set the bitfields corresponding to actual supported operations on the resource and the bits corresponding to `AclOperation.ALL`, `AclOperation.ANY` and `AclOperation.UNKNOWN` will never be set by the broker.

Metadata Request and Response: v8

Metadata v8

```
Metadata Request => [topics] allow_auto_topic_creation include_cluster_authorized_operations
include_topic_authorized_operations <== ADDED include_cluster_authorized_operations,
include_topic_authorized_operations
  topics => STRING
  allow_auto_topic_creation => BOOLEAN
  include_cluster_authorized_operations => BOOLEAN <== NEW
  include_topic_authorized_operations => BOOLEAN <== NEW

Metadata Response => throttle_time_ms [brokers] cluster_id controller_id [topic_metadata] authorized_operations
<== ADDED authorized_operations
  throttle_time_ms => INT32
  brokers => node_id host port rack
    node_id => INT32
    host => STRING
    port => INT32
    rack => NULLABLE_STRING
  cluster_id => NULLABLE_STRING
  controller_id => INT32
  topic_metadata => error_code topic is_internal [partition_metadata] authorized_operations <== ADDED
  authorized_operations
    error_code => INT16
    topic => STRING
    is_internal => BOOLEAN
    partition_metadata => error_code partition leader leader_epoch [replicas] [isr] [offline_replicas]
      error_code => INT16
      partition => INT32
      leader => INT32
      leader_epoch => INT32
      replicas => INT32
      isr => INT32
      offline_replicas => INT32
    authorized_operations => INT32 <== NEW
  authorized_operations => INT32 <== NEW
```

DescribeGroups Request and Response v3

DescribeGroups v3

```
DescribeGroups Request => [group_ids] include_authorized_operations <== ADDED include_authorized_operations
group_ids => STRING
include_authorized_operations => BOOLEAN <== NEW

DescribeGroups Response => throttle_time_ms [groups]
  throttle_time_ms => INT32
  groups => error_code group_id state protocol_type protocol [members] authorized_operations <== ADDED
  authorized_operations
    error_code => INT16
    group_id => STRING
    state => STRING
    protocol_type => STRING
    protocol => STRING
    members => member_id client_id client_host member_metadata member_assignment
      member_id => STRING
      client_id => STRING
      client_host => STRING
      member_metadata => BYTES
      member_assignment => BYTES
    authorized_operations => INT32 <== NEW
```

AdminClient API Changes

All relevant `DescribeXxxOptions` classes will include a new field and corresponding accessors to request authorized operations. By default the option is disabled. The classes affected are:

- `DescribeClusterOptions`: This is a metadata request. Operations returned will be a subset of {Describe, Alter, DescribeConfigs, AlterConfigs, IdempotentWrite, ClusterAction, Create}
- `DescribeTopicsOptions`: Also a metadata request. Operations returned will be a subset of {Create, Delete, Read, Write, Describe, Alter, DescribeConfigs, AlterConfigs}
- `DescribeConsumerGroupsOptions`: Operations returned will be a subset of {Describe, Read, Delete}

Example: DescribeTopicsOptions

DescribeTopicOptions

```
public class DescribeTopicsOptions extends AbstractOptions<DescribeTopicsOptions> {

    private boolean includeAuthorizedOperations;
    /** Retain existing code here */

    public DescribeTopicsOptions includeAuthorizedOperations(boolean includeAuthorizedOperations) {
        this.includeAuthorizedOperations = includeAuthorizedOperations;
        return this;
    }

    public boolean includeAuthorizedOperations() {
        return includeAuthorizedOperations;
    }
}
```

The corresponding resource description or result classes returned by AdminClient will be extended to provide an optional set of authorized operations. The bits set in the new INT32 field in the response will be used to generate a set of `AclOperation` entries, where each operation code is derived from the position of the bit field. When new operations are added, older clients will ignore any authorized operations returned by the broker that is not supported by the client. The returned `AclOperation` set will never contain `AclOperation.ANY`, `AclOperation.ALL` or `AclOperation.UNKNOWN`.

Example: ConsumerGroupDescription

ConsumerGroupDescription

```
public class ConsumerGroupDescription {
    private Set<AclOperation> authorizedOperations;
    /** Retain existing code here */

    public Set<AclOperation> authorizedOperations() {
        if (authorizedOperations.isEmpty())
            throw new IllegalArgumentException("Authorized operations were not provided by the broker");
        return authorizedOperations;
    }
}
```

ResourceType API Changes

A new method will be added to the `kafka.security.auth.ResourceType` trait to obtain the supported operations associated with a resource type. This will be used to maintain supported operations for a resourceType.

This can be used by custom authorizers to determine authorized operations.

ResourceType changes

```
Sealed trait ResourceType extends BaseEnum with Ordered[ ResourceType ] {
  ....

  // this method output will not include "All" Operation type
  def supportedOperations: Set[Operation]
}

case object Topic extends ResourceType {
  ...
  val supportedOperations = Set(Read, Write, Create, Describe, Delete, Alter, DescribeConfigs, AlterConfigs)
}

case object Group extends ResourceType {
  ...
  val supportedOperations = Set(Read, Describe, Delete)
}

case object Cluster extends ResourceType {
  ...
  val supportedOperations = Set(Create, ClusterAction, DescribeConfigs, AlterConfigs, IdempotentWrite, Alter, Describe)
}

case object TransactionalId extends ResourceType {
  ...
  val supportedOperations = Set(Describe, Write)
}

case object DelegationToken extends ResourceType {
  ...
  val supportedOperations = Set(Describe)
}
```

Proposed Changes

As described above, Kafka protocol for requests and responses to describe broker resources will be extended to request authorized operations and return the set of authorized operations if requested. Broker will use its pluggable `Authorizer` to obtain the set of permitted operations for the `Session` performing the `Describe` operation. If no authorizer is configured on the broker, the full set of supported operations on each resource will be returned.

Broker will check `Describe` access on the resources before returning any metadata, so only users authorized for `Describe` may obtain the additional information provided by this KIP. Users without `Describe` access continue to get errors that don't leak information about the existence of resources.

Compatibility, Deprecation, and Migration Plan

- Existing clients using older versions will not request authorized operations in `Describe` requests since the default is to disable this feature. This keeps older clients compatible with newer brokers.
- Newer clients connecting to older brokers will use the older protocol version and hence will not request authorized operations.
- When the *AdminClient* is talking to a broker which does not support KIP-430, it will fill in either null or *UnsupportedVersionException* for the returned ACL operations fields in objects. For example, *ConsumerGroupDescription#authorizedOperations* will be null if the broker did not supply this information. *DescribeClusterResult#authorizedOperations* will throw an *UnsupportedVersionException* if the broker did not supply this information.
- When new operations are added, newer brokers may return operations that are not known to older clients. AdminClient will ignore any bit that is set in `authorized_operations` that is not known to the client. The `Set<AclOperation>` created by the client from the bits returned by the broker will only include operations that the client knows about.

Changes dropped from the Proposal

Initially we proposed below API changes to the scala `Authorizer` trait with a default implementation so that existing implementations continue to work. But Scala 2.11 doesn't convert the default implementation in a trait to a default implementation in Java. So this breaks existing Java authorizer implementations when building with scala 2.11 version of core. Due to this we dropped below Authorizer API related changes. These changes can be implemented in future when we drop support for Scala 2.11.

Authorizer API Changes

A new method will be added to the `kafka.security.auth.Authorizer` interface to obtain the collection of authorized operations associated with a resource. Default implementation of this method will use the existing `authorize()` API to check every supported operation on the resource. This ensures that custom authorizers will continue to work without change. The built-in authorizer implementation `SimpleAclAuthorizer` will include a more performant implementation that traverses ACLs once to retrieve all the authorized operations for the user. All permitted operations on the resource including any that are implicitly allowed by ACLs will be included in the returned set. For example, if a `Read` ACL is found, both `Read` and `Describe` will be included since both are permitted. If an ACL is found for all operations (`AclOperation.ALL`) on a resource, broker will explicitly list all supported operations of the resource, so that clients always receive the full set of actual permitted operations.

Authorizer API changes

```
trait Authorizer extends Configurable {  
  def authorizedOperations(session: Session, resource: Resource): Set[Operation] = {  
    // Use authorize() to obtain permitted operations for the `session` on `resource`  
  }  
  ....  
}
```

Rejected Alternatives

Add a new request to obtain authorized operations for different resources

Instead of extending `Describe` requests and responses, we could add a new request to determine authorized operations on a resource or set of resources. This KIP proposes to extend `Describe` responses since that makes it easier for clients to obtain all the metadata using a single request. It also makes `Describe` responses self-contained with all the relevant information together in a single response and avoids clients having to deal with changes to resources or ACLs while the results are combined from separate responses. Since access to this information is controlled using `Describe` ACLs anyway, this extension is consistent with our security model and does not leak any information that cannot currently be obtained by users with a specific set of ACLs.