

KIP-518: Allow listing consumer groups per state

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [ListGroup API](#)
 - [AdminClient API](#)
 - [Command line tools](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

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

At the moment, the ListGroups API always returns all groups. There are metrics that track the number of groups in several states but the only way to know the state of a group is to describe it using the DescribeGroups API. In large clusters with hundreds or thousands of groups, it can be hard for a cluster administrator to identify which groups are stable or which groups are dead and can be deleted for example. Basically this requires describing a large number of groups which is inefficient in terms of time for the administrator and load for the cluster.

To improve that use case, I propose to allow listing groups by state. For example listing stable groups allows to immediately tell if several applications are healthy if their groups are all returned without needing to describe each group. This would allow building tools and user interfaces that can easily and quickly display informations about live groups.

Public Interfaces

ListGroups API

Bump ListGroupsRequest version to 4 and include a new field "StatesFilter".

```
{
  "apiKey": 16,
  "type": "request",
  "name": "ListGroupsRequest",
  // Version 1 and 2 are the same as version 0.
  //
  // Version 3 is the first flexible version.
  //
  // Version 4 adds the StatesFilter field (KIP-518).
  "validVersions": "0-4",
  "flexibleVersions": "3+",
  "fields": [
    { "name": "StatesFilter", "type": "[[]string", "versions": "4+",
      "about": "The states of the groups we want to list. If empty all groups are returned with their state."
    }
  ]
}
```

Bump ListGroupsResponse version to 4 and include a new field "GroupState".

```

{
  "apiKey": 16,
  "type": "response",
  "name": "ListGroupResponse",
  // Version 1 adds the throttle time.
  //
  // Starting in version 2, on quota violation, brokers send out responses before throttling.
  //
  // Version 3 is the first flexible version.
  //
  // Version 4 adds the GroupState field (KIP-518).
  "validVersions": "0-4",
  "flexibleVersions": "3+",
  "fields": [
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "1+", "ignorable": true,
      "about": "The duration in milliseconds for which the request was throttled due to a quota violation, or
zero if the request did not violate any quota." },
    { "name": "ErrorCode", "type": "int16", "versions": "0+",
      "about": "The error code, or 0 if there was no error." },
    { "name": "Groups", "type": "[[]ListedGroup", "versions": "0+",
      "about": "Each group in the response.", "fields": [
        { "name": "GroupId", "type": "string", "versions": "0+", "entityType": "groupId",
          "about": "The group ID." },
        { "name": "ProtocolType", "type": "string", "versions": "0+",
          "about": "The group protocol type." },
        { "name": "GroupState", "type": "string", "versions": "4+", "ignorable": true,
          "about": "The group state name." }
      ]
    }
  ]
}

```

This KIP bumps the version of this request to avoid any version incompatibility issues with older broker. See rejected alternatives.

If the version is 4 or above, if `StatesFilter` is empty in the request all groups will be returned with their states. If `StatesFilter` is set to a list of group states, only states that are currently in these states will be returned. At the moment, consumer group state is already serialized as a `STRING` in `DescribeGroupsResponse`, so serializing it the same way here.

When attempting to list groups with states using an older broker, an `UnsupportedVersionException` will be raised in the client.

AdminClient API

To expose this feature in the AdminClient API, `ListConsumerGroupsOptions` will be updated.

ListConsumerGroupsOptions

```

public class ListConsumerGroupsOptions extends AbstractOptions<ListConsumerGroupsOptions> {

    /**
     * Only groups in these states will be returned by listConsumerGroups()
     * If not set, all groups are returned without their states
     * throw IllegalArgumentException if states is empty
     */
    public ListConsumerGroupsOptions inStates(Set<ConsumerGroupState> states) { }

    /**
     * Returns the list of States that are requested
     */
    public Optional<Set<ConsumerGroupState>> states() { }
}

```

Similarly, a state field will be exposed in `ConsumerGroupListing`:

```

public class ConsumerGroupListing {

    /**
     * Consumer Group state
     */
    public Optional<ConsumerGroupState> state() {
        return state;
    }
}

```

The state will be empty when ListGroupsRequest < 3 is used, for example when connecting to an older broker.

Command line tools

To expose this feature in the command line tools, ConsumerGroupCommand tool will be updated to support filtering by group states. The existing flag "--state" will be updated to allow specifying a list of group states when used with "--list". When specified, the state of each group will be printed along the group id.

For example:

```

# Existing behaviour
./kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
group1
group2

# Listing all groups with their states
./kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list --state
GROUP          STATE
group1         stable
group2         Empty

# Listing groups in the stable state
./kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list --state Stable
GROUP          STATE
group1         Stable

```

Compatibility, Deprecation, and Migration Plan

With this KIP, having Describe authorization on the Cluster resource allows to retrieve the state of all groups. Previously this required having Describe authorization on each Group.

Rejected Alternatives

- Allow specifying a filter in ListGroups to retrieve groups by name or by state using wildcard
 - Unclear if it's a common use case
- Add flexible fields without bumping the ListGroups version
 - Keeping v3 would allow cases where brokers could support v3 but not the new optional field. In that case, if a client specifies states in its request, brokers would still return the full list of groups without states. This looked confusing. In case broker don't support a feature, a UnsupportedOperationException is preferred.