KIP-670: Add ConsumerGroupCommand to delete static members

Status

Current state: *Under Discussion*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *https://issues.apache.org/jira/browse/KAFKA-9440*

Motivation

We introduced a new AdminClient API removeMembersFromConsumerGroup in 2.4. It would be good to instantiate the API as part of the ConsumerGroupCommand for easy command line usage.

We can add new option (--remove-members) to remove members from group in ConsumerGroupCommand. Users can either remove member by providing member id (--member) or can delete all the members of group using (--all-members) option.

We can use exiting option (–-group) to accept groupId from CLI

Existing API of org/apache/kafka/clients/admin/Admin.java can be used for removing all members.

```
/**
 * Remove members from the consumer group by given member identities.
 * <p>
 * For possible error codes, refer to {@link LeaveGroupResponse}.
 *
 * @param groupId The ID of the group to remove member from.
 * @param options The options to carry removing members' information.
 * @return The MembershipChangeResult.
 */
RemoveMembersFromConsumerGroupResult removeMembersFromConsumerGroup(String groupId,
RemoveMembersFromConsumerGroupOptions options);
```

For removing selected members we can introduce new API in same interface which will accepts memberIds to remove only selected members from consumer groups.

```
/**
 * Remove members from the consumer group by given member identities.
 * <p>
 * For possible error codes, refer to {@link LeaveGroupResponse}.
 *
 * @param groupId The ID of the group to remove member from.
 * @param memberIds The groupInstanceIds of the members te be removed.
 * @return The MembershipChangeResult.
 */
RemoveMembersFromConsumerGroupResult removeMembersFromConsumerGroup(String groupId, Collection<String>
memberIds);
```

Proposed Changes

Implementation of newly created API method will delegate the call to existing API after building remove member options object from given memberIds

```
@Override
public RemoveMembersFromConsumerGroupResult removeMembersFromConsumerGroup(String groupId, Collection<String>
memberIds) {
        Set<MemberToRemove> members = new HashSet<>();
        memberIds.forEach(memberId -> {
                members.add(new MemberToRemove(memberId));
        });
        RemoveMembersFromConsumerGroupOptions options = new RemoveMembersFromConsumerGroupOptions(members);
        return removeMembersFromConsumerGroup(groupId, options);
}
```

Implementation of delegating call to KafkaAdminClient from kafka/admin/ConsumerGroupCommand.scala may look like this :

```
 def deleteMembersFromConsumerGroup(): Boolean = {
      val groupId = opts.options.valueOf(opts.groupOpt)
      val result = if (opts.options.has(opts.allMembersOpts)) {
        adminClient.removeMembersFromConsumerGroup(groupId, new RemoveMembersFromConsumerGroupOptions)
      } else {
        val memberIds = opts.options.valuesOf(opts.memberOpt).asScala
        adminClient.removeMembersFromConsumerGroup(groupId, memberIds.asJava)
      }

      try {
        result.all.get
        println("Deletion of requested member(s) of consumer group was successful.")
        return true
      } catch {
        case e: Throwable => {
          printError(s"Deletion of requested member(s) of consumer group failed due to ${e.getMessage}")
          return false
        }
      }
      false
    }
```

Compatibility, Deprecation, and Migration Plan

No users will be impacted as this is just addition of two new methods.

Rejected Alternatives

None