KIP-351: Add --under-min-isr option to describe topics command

Status

Current state: ACCEPTED

Discussion thread: *here*

Previous Discussion thread: here

Vote thread: here

JIRA: KAFKA-7236

PR: https://github.com/apache/kafka/pull/6224

Motivation

The "min.insync.replicas" configuration specifies the minimum number of insync replicas required for a partition to accept messages from the producer. If the insync replica count of a partition falls under the specified "min.insync.replicas", then the broker will reject messages for producers using acks=all. These producers will suffer unavailability as they will see a NotEnoughReplicas or NotEnoughReplicasAfterAppend exception.

We currently have an UnderMinIsrPartitionCount metric which is useful for identifying when partitions fall under "min.insync.replicas", however it is still difficult to identify which topic partitions are affected and need fixing.

We can leverage the describe topics command in TopicCommand to add an option "--under-min-isr-partitions" to list out exactly which topic partitions are below "min.insync.replicas" and need fixing to maintain availability.

Public Interfaces

This change would add an additional flag "--under-min-isr-partitions" to TopicCommand, but the output will follow the same format as the "under-replicated-partitions" and "offline-partitions" options.

```
val reportUnderMinIsrPartitionsOpt = parser.accepts("under-min-isr-partitions",
                                                    "if set when describing topics,
only show partitions which are under the configured minimum in-sync replica count")
```

Proposed Changes

The challenge with supporting this additional feature is that the "min.insync.replicas" configuration may be set at a broker or topic level.

We can use the `AdminClient.describeConfigs` on the topics as that the API call will give us the "computed" proper values for configurations (ConfigSource as "DYNAMIC_TOPIC_CONFIG", "DYNAMIC_BROKER_CONFIG", "DYNAMIC_DEFAULT_BROKER_CONFIG", "STATIC_BROKER_CONFIG", and "DEFAULT_CONFIG").

We can pre-fetch the "computed" topic configurations if "--under-min-isr-partitions" option is specified to avoid making a separate AdminClient call per topic.

```
# Assuming we have an AdminClient instance
val adminClient = ...

// Pre-fetch and get "computed" topic configs for all specified topics
val computedTopicConfigs = if (reportUnderMinIsrPartitions)
  Option(adminClient.describeConfigs(
    topics.map(topic => new ConfigResource(ConfigResource.Type.TOPIC, topic)).asJavaCollection).all().get())
else None

for (topic <- topics)
  ...
  if (describePartitions) {
    // Get "computed" topic "min.insync.replicas" for this topic
    val computedTopicMinISR = if (reportUnderMinISRPartitions)
      Option(computedTopicConfigs.get.get(new ConfigResource(ConfigResource.Type.TOPIC, topic))
      .get(TopicConfig.MIN_IN_SYNC_REPLICAS_CONFIG).value().toInt) else None

    for ((partitionId, assignedReplicas) <- sortedPartitions) {
      ...

      // Print current topic partition if reportUnderMinISRPartitions and ISR count < "computed" min ISR
      if (... ||
        (reportUnderMinISRPartitions && inSyncReplicas.size < computedTopicMinISR.get) {
      ...
```

This means we need an additional flag "--bootstrap-server" to use AdminClient. KIP-377: TopicCommand to use AdminClient is already proposing a change to use AdminClient and introduce a "--bootstrap-server" option, so we can leverage the changes in KIP-377 for this KIP.

NOTE: This option is not supported with the deprecated "--zookeeper" option.

Compatibility, Deprecation, and Migration Plan

Since we rely on AdminClient for the computed "min.insync.replicas" configuration, this new option **CANNOT be used with the deprecated "--zookeeper" option.**

Rejected Alternatives

None so far.