KIP-548 Add Option to enforce rack-aware custom partition reassignment execution

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA:** **KAFKA-9205

Motivation

One regularly used healing operation on Kafka clusters is replica reassignments for topic partitions. For example, when there is a skew in inbound /outbound traffic of a broker replica reassignment can be used to move some leaders/followers from the broker; or if there is a skew in disk usage of brokers, replica reassignment can more some partitions to other brokers that have more disk space available.

In Kafka clusters that span across multiple data centers (or availability zones), high availability is a priority; in the sense that when a data center goes offline the cluster should be able to resume normal operation by guaranteeing partition replicas in all data centers.

This guarantee is currently the responsibility of the on-call engineer that performs the reassignment or the tool that automatically generates the reassignment plan for improving the cluster health (e.g. by considering the rack configuration value of each broker in the cluster). the former, is quite error-prone, and the latter, would lead to duplicate code in all such admin tools (which are not error free either). Not all use cases can make use the default assignment strategy that is used by --generate option; and current rack aware enforcement applies to this option only.

It would be great for the built-in replica assignment API and tool provided by Kafka to support a rack aware verification option for **--execute** scenario that would simply return an error when [some] brokers in any replica set share a common rack.

Public Interfaces

Existing API will be updated under Admin tools

```
def executeAssignment(zkClient: KafkaZkClient, adminClientOpt: Option[JAdminClient], opts:
ReassignPartitionsCommandOptions)
```

executeAssignment() will by default verify rack awareness into consideration when re-assigning the partitions.

*Rack awareness should be explicitly disable using '--disable-rack-aware' option when executing **--execute** command.*

For example:

```
./kafka-reassign-partitions.sh --reassignment-json-file partitions-to-move.json --execute --zookeeper localhost:
2181 --disable-rack-aware

$ cat partitions-to-move.json
{
  "partitions": [
    {
      "topic": "foo",
      "partition": 6,
      "replicas": [
        1004,
        1003,
        1005
      ],
      "log_dirs": [
        "any",
        "any",
        "any"
      ]
    }
  ],
  "version": 1
}
```

Proposed Changes

Before this proposal:

If kafka-reassign-partitions is used with custom reassignment algorithm or if the reassignment is manually generated on ad-hoc basic, the tool does not enforce rack awareness when run with --execute option.

After this proposal:

The --execute command by default take rack awareness into consideration, and if the custom generated reassignment planner has conflicts along with the racks then it will throw the error msg with appropriate reason and conflict of partitions along with the racks info. The users need to explicitly choose the option --disable-rack-aware if they want to ignore the rack awareness.

Compatibility, Deprecation, and Migration Plan

This gonna change the current behavior of --execute command, because if the reassignment planner is generated by a custom script or ad-hoc (not by --generate option), then there are high chances of conflicts with rack awareness. So, in-order to pass through the errors, users have two options,

1. Users can set the --disable-rack-aware option to ignore the rack awareness and pass through the errors. or
2. Users need to generate the reassignment planner using --generate option (no custom planner).

By this change the usage of options in --execute command will be aligned with --generate option, the rack awareness will be consider by default both for --generate as well as --execute unless explicitly set to --disable-rack-aware.

There is not deprecation or migration plan needed for this change.

Rejected Alternatives

None.