

# MEP 7 -- Watch&Release channel via etcd

Current state: Under Discussion

ISSUE: <https://github.com/milvus-io/milvus/issues/6578>

PRs:

Keywords: etcd, datacoord, datanode

Released:

## Summary

DataCoord register channels on etcd and DataNode watch etcd to do watch/release operations.

## Motivation

There are several problems when DataCoord sends the WatchDmChannel to the DataNode through grpc:

1. If datacoord cannot connect to datanode, it needs to try again. Retrial failure requires reallocation, which may result in duplicate watches.
2. If datacoord has load balance, it needs to send unwatch and watch request, which may also lead to failure and retrying.

## Public Interfaces

Remove WatchDmChannel of DataNode.

## Design Details

Etcd keychannel / [nodeID] / [channelName]value: ChannelInfo

ChannelInfo contains State,StartTime, VchannelInfo

1. State is a enum whose values are Unwatched, Watched. This means whether datanode watch it successfully.
2. StartTime is the watch event start time.
3. VchannelInfo contains all info needed to restore the channel.

If there is a new channel registration, datacoord updates channel / [nodeid] / [channelname]

Datanode monitors the ADD and DELETE events of channel / [nodeid]

DataCoord

1. When the datacoord is started, the channels of offline datanodes are assigned to current online nodes.
2. When DataNode comes online, DataCoord may move some channels to the node and change the channels of different nodes through etcd transactions operation.
3. When DataNode goes offline, DataCoord reassigns the channels to other nodes, changing them through the etcd transaction.
4. Specially, if the last DataNode goes offline and there is no living DataNode at this time, record the channel in channel/remaining[channelName].
5. Start a background goroutine to check states of channels. If a channel's state has't changed to Watched for a long time, maybe we should reallocated it to another node atomically.

DataNode:

1. When DataNode starts, the channel of this Node on etcd must be empty, because the nodeID is incremented.
2. When DataNode receives an Add event, execute WatchChannel, and transactionly change state of channel on etcd to Watched.
3. When DataNode receives Delete event, execute ReleaseChannel.

## Test Plan