CEPHFS HEALTH MESSAGES

CLUSTER HEALTH CHECKS

The Ceph monitor daemons will generate health messages in response to certain states of the filesystem map structure (and the enclosed MDS maps).

Message: mds rank(s) *ranks* have failed Description: One or more MDS ranks are not currently assigned to an MDS daemon; the cluster will not recover until a suitable replacement daemon starts.

Message: mds rank(s) *ranks* are damaged Description: One or more MDS ranks has encountered severe damage to its stored metadata, and cannot start again until it is repaired.

Message: mds cluster is degraded Description: One or more MDS ranks are not currently up and running, clients may pause metadata IO until this situation is resolved. This includes ranks being failed or damaged, and additionally includes ranks which are running on an MDS but have not yet made it to the *active* state (e.g. ranks currently in *replay* state).

Message: mds *names* are laggy Description: The named MDS daemons have failed to send beacon messages to the monitor for at least `mds_beacon_grace` (default 15s), while they are supposed to send beacon messages every `mds_beacon_interval` (default 4s). The daemons may have crashed. The Ceph monitor will automatically replace laggy daemons with standbys if any are available.

Message: insufficient standby daemons available Description: One or more file systems are configured to have a certain number of standby daemons available (including daemons in standby-replay) but the cluster does not have enough standby daemons. The standby deamons not in replay count towards any file system (i.e. they may overlap). This warning can configured by setting `ceph fs set <fs> standby_count_wanted <count>`. Use zero for `count` to disable.

DAEMON-REPORTED HEALTH CHECKS

MDS daemons can identify a variety of unwanted conditions, and indicate these to the operator in the output of `ceph status`. This conditions have human readable messages, and additionally a unique code starting MDS_HEALTH which appears in JSON output.

Message: "Behind on trimming…" Code: MDS_HEALTH_TRIM Description: CephFS maintains a metadata journal that is divided into *log segments*. The length of journal (in number of segments) is controlled by the setting `mds_log_max_segments`, and when the number of segments exceeds that setting the MDS starts writing back metadata so that it can remove (trim) the oldest segments. If this writeback is happening too slowly, or a software bug is preventing trimming, then this health message may appear. The threshold for this message to appear is for the number of segments to be double `mds_log_max_segments`.

Message: "Client *name* failing to respond to capability release" Code: MDS_HEALTH_CLIENT_LATE_RELEASE, MDS_HEALTH_CLIENT_LATE_RELEASE_MANY Description: CephFS clients are issued *capabilities* by the MDS, which are like locks. Sometimes, for example when another client needs access, the MDS will request clients release their capabilities. If the client is unresponsive or buggy, it might fail to do so promptly or fail to do so at all. This message appears if a client has taken longer than `session_timeout` (default 60s) to comply.

Message: "Client *name* failing to respond to cache pressure" Code: MDS_HEALTH_CLIENT_RECALL, MDS_HEALTH_CLIENT_RECALL_MANY Description: Clients maintain a metadata cache. Items (such as inodes) in the client cache are also pinned in the MDS cache, so when the MDS needs to shrink its cache (to stay within `mds_cache_size` or `mds_cache_memory_limit`), it sends messages to clients to shrink their caches too. If the client is unresponsive or buggy, this can prevent the MDS from properly staying within its cache limits and it may eventually run out of memory and crash. This message appears if a client has taken more than `mds_recall_state_timeout` (default 60s) to comply.

Message: "Client *name* failing to advance its oldest client/flush tid" Code: MDS_HEALTH_CLIENT_OLDEST_TID, MDS_HEALTH_CLIENT_OLDEST_TID_MANY Description: The CephFS client-MDS protocol uses a field called the *oldest tid* to inform the MDS of which client requests are fully complete and may therefore be forgotten about by the MDS. If a buggy client is failing to advance this field, then the MDS may be prevented from properly cleaning up resources used by client requests. This message appears if a client appears to have more than `max_completed_requests` (default 100000) requests that are complete on the MDS side but haven't yet been accounted for in the client's *oldest tid* value.

Message: "Metadata damage detected" Code: MDS_HEALTH_DAMAGE, Description: Corrupt or missing metadata was encountered when reading from the metadata pool. This message indicates that the damage was sufficiently isolated for the MDS to continue operating, although client accesses to the damaged subtree will return IO errors. Use the `damage ls` admin socket command to get more detail on the damage. This message appears as soon as any damage is encountered.

Message: “MDS in read-only mode” Code: MDS_HEALTH_READ_ONLY, Description: The MDS has gone into readonly mode and will return EROFS error codes to client operations that attempt to modify any metadata. The MDS will go into readonly mode if it encounters a write error while writing to the metadata pool, or if forced to by an administrator using the *force_readonly* admin socket command.

Message: *N* slow requests are blocked” Code: MDS_HEALTH_SLOW_REQUEST, Description: One or more client requests have not been completed promptly, indicating that the MDS is either running very slowly, or that the RADOS cluster is not acknowledging journal writes promptly, or that there is a bug. Use the ops admin socket command to list outstanding metadata operations. This message appears if any client requests have taken longer than `mds_op_complaint_time` (default 30s).

Message: “Too many inodes in cache” Code: MDS_HEALTH_CACHE_OVERSIZED Description: The MDS is not succeeding in trimming its cache to comply with the limit set by the administrator. If the MDS cache becomes too large, the daemon may exhaust available memory and crash. By default, this message appears if the actual cache size (in inodes or memory) is at least 50% greater than `mds_cache_size` (default 100000) or `mds_cache_memory_limit` (default 1GB). Modify `mds_health_cache_threshold` to set the warning ratio.