‹cbstats›

Retrieve lower level statistics with the ‹cbstats› utility, which provides deep insight into what occurs within a cluster.

Syntax

The basic syntax is:

```
cbstats [host]: ‹port› [command] [options]
```

Description

The ‹cbstats› tool queries per-node and per-bucket. The IP address of a node in the cluster and a named bucket must be specified.

If you don't specify a bucket (using the option ‹-b›), you will get information for the ‹‹default›› bucket, if it exists. If the ‹‹default›› bucket does not exist, the output will be empty.

Warning: Do not use the legacy memcached port 11211 when using ‹cbstats› since this port won't be available after server-side moxi is removed in the next release.

The tool is found in the following locations:

colspecs...

Platform	Location
Linux	‹/opt/couchbase/bin/cbstats›
Windows	‹C:\Program Files\Couchbase\Server\bin\cbstats.exe›
Mac OS X	‹ /Applications/Couchbase Server.app/Contents/Resources/cou core/bin/cbstats›

This following table shows top-level statistics:

Table 1: Top-level statistics

colspecs...

Stat	Description
uuid	The unique identifier for the bucket. It is specified per bucket, therefore you must specify the bucket you are interested in.
ep_version	Version number of ep_engine.
ep_storage_age	Seconds since most recently stored object was initially queued.
ep_storage_age_highwat	ep_storage_age high water mark.
ep_startup_time	System-generated engine startup time.
ep_data_age	Seconds since most recently stored object was modified.
ep_data_age_highwat	ep_data_age high water mark.
ep_too_young	Number of times an object was not stored due to being too young.
ep_too_old	Number of times an object was stored after being dirty too long.
ep_total_enqueued	Total number of items queued for persistence.
ep_total_new_items	Total number of persisted new items.
ep_total_del_items	Total number of persisted deletions.

ep_total_persisted	Total number of items persisted.
ep_item_flush_failed	Number of times an item failed to flush due to storage errors.
ep_item_commit_failed	Number of times a transaction failed to commit due to storage errors.
ep_item_begin_failed	Number of times a transaction failed to start due to storage errors.
ep_expired_access	Number of times an item was expired on application access.
ep_expired_pager	Number of times an item was expired by ep engine item pager.
ep_item_flush_expired	Number of times an item is not flushed due to the expiry of the item.
ep_queue_size	Number of items queued for storage.
ep_flusher_todo	Number of items currently being written.
ep_flusher_state	Current state of the flusher thread.
ep_commit_num	Total number of write commits.
ep_commit_time	Number of milliseconds of most recent commit.
ep_commit_time_total	Cumulative milliseconds spent committing.
ep_vbucket_del	Number of vbucket deletion events.
ep_vbucket_del_fail	Number of failed vbucket deletion events.
ep_vbucket_del_max_walltime	Max wall time (μs) spent by deleting a vbucket.
ep_vbucket_del_avg_walltime	Avg wall time (μs) spent by deleting a vbucket.
ep_flush_duration_total	Cumulative seconds spent flushing.
ep_flush_all	True if disk flush_all is scheduled.
ep_num_ops_get_meta	Number of getMeta operations.
ep_num_ops_set_meta	Number of setWithMeta operations.
ep_num_ops_del_meta	Number of delWithMeta operations..
ep_num_ops_set_meta_res_failed	Number of setWithMeta ops that failed conflict resolution.
ep_num_ops_del_meta_res_failed	Number of delWithMeta ops that failed conflict resolution.
ep_num_ops_set_ret_meta	Number of setRetMeta operations.
ep_num_ops_del_ret_meta	Number of delRetMeta operations.
curr_items	Num items in active vbuckets (temp + live).
curr_temp_items	Num temp items in active vbuckets.
curr_items_tot	Num current items including those not active (replica, dead and pending states).
ep_kv_size	Memory used to store item metadata, keys and values, no matter the vbucket's state. If an item's value is ejected, this stats will be decremented by the size of the item's value.
ep_value_size	Memory used to store values for resident keys
ep_overhead	Extra memory used by transient data like persistence queues, replication queues, checkpoints, etc.
ep_mem_low_wat	Low water mark for auto-evictions.
ep_mem_high_wat	High water mark for auto-evictions.
ep_total_cache_size	The total byte size of all items, no matter the vbucket's state, no matter if an item's value is ejected.
ep_oom_errors	Number of times unrecoverable OOMs happened while processing operations.
ep_tmp_oom_errors	Number of times temporary OOMs happened while processing operations.

ep_mem_tracker_enabled	True if memory usage tracker is enabled.
ep_bg_fetched	Number of items fetched from disk.
ep_bg_meta_fetched	Number of meta items fetched from disk.
ep_bg_remaining_jobs	Number of remaining bg fetch jobs.
ep_max_bg_remaining_jobs	Max number of remaining bg fetch jobs that we have seen in the queue so far.
ep_tap_bg_fetched	Number of tap disk fetches.
ep_tap_bg_fetch_requeued	Number of times a tap bg fetch task is requeued.
ep_num_pager_runs	Number of times we ran pager loops to seek additional memory.
ep_num_expiry_pager_runs	Number of times we ran expiry pager loops to purge expired items from memory/disk.
ep_num_access_scanner_runs	Number of times we ran access scanner to snapshot working set.
ep_access_scanner_num_items	Number of items that last access scanner task swept to access log.
ep_access_scanner_task_time	Time of the next access scanner task (GMT).
ep_access_scanner_last_runtime	Number of seconds that last access scanner task took to complete.
ep_items_rm_from_checkpoints	Number of items removed from closed unreferenced checkpoints.
ep_num_value_ejects	Number of times item values got ejected from memory to disk.
ep_num_eject_failures	Number of items that could not be ejected.
ep_num_not_my_vbuckets	Number of times `Not My vBucket` exception happened during runtime.
ep_tap_keepalive	Tap keepalive time.
ep_dbname	DB path.
ep_io_num_read	Number of I/O read operations.
ep_io_num_write	Number of I/O write operations.
ep_io_read_bytes	Number of bytes read (key + values).
ep_io_write_bytes	Number of bytes written (key + values).
ep_pending_ops	Number of ops awaiting pending vbuckets.
ep_pending_ops_total	Total blocked pending ops since reset.
ep_pending_ops_max	Max ops seen awaiting 1 pending vbucket.
ep_pending_ops_max_duration	Max time (μs) used waiting on pending vbuckets.
ep_bg_num_samples	The number of samples included in the average.
ep_bg_min_wait	The shortest time (μs) in the wait queue.
ep_bg_max_wait	The longest time (μs) in the wait queue.
ep_bg_wait_avg	The average wait time (μs) for an item before it's serviced by the dispatcher.
ep_bg_min_load	The shortest load time (μs).
ep_bg_max_load	The longest load time (μs).
ep_bg_load_avg	The average time (μs) for an item to be loaded from the persistence layer.
ep_num_non_resident	The number of non-resident items.
ep_bg_wait	The total elapse time for the wait queue.
ep_bg_load	The total elapse time for items to be loaded from the persistence layer.
ep_allow_data_loss_during_shutdown	Whether data loss is allowed during server shutdown.
ep_alog_block_size	Access log block size.

ep_alog_path	Path to the access log.
ep_alog_sleep_time	Interval between access scanner runs in minutes.
ep_alog_task_time	Hour in GMT time when access scanner task is scheduled to run.
ep_backend	The backend that is being used for data persistence.
ep_bg_fetch_delay	The amount of time to wait before doing a background fetch.
ep_chk_max_items	The number of items allowed in a ‹checkpoint› before a new one is created.
ep_chk_period	The maximum lifetime of a checkpoint before a new one is created.
ep_chk_persistence_remains	Number of remaining vbuckets for checkpoint persistence.
ep_chk_persistence_timeout	Timeout for vbucket checkpoint persistence.
ep_chk_remover_stime	The time interval for purging closed checkpoints from memory.
ep_config_file	The location of the ep-engine config file.
ep_couch_bucket	The name of this bucket.
ep_couch_host	The hostname that the CouchDB views server is listening on.
ep_couch_port	The port the CouchDB views server is listening on.
ep_couch_reconnect_sleeptime	The amount of time to wait before reconnecting to CouchDB.
ep_couch_response_timeout	Length of time to wait for a response from CouchDB before reconnecting.
ep_data_traffic_enabled	Whether or not data traffic is enabled for this bucket.
ep_degraded_mode	True if the engine is either warming up or data traffic is disabled.
ep_exp_pager_stime	The time interval for purging expired items from memory.
ep_failpartialwarmup	True if we want kill the bucket if warmup fails.
ep_flushall_enabled	True if this bucket enables the use of the flush_all command.
ep_getl_default_timeout	The default getl lock duration.
ep_getl_max_timeout	The maximum getl lock duration.
ep_ht_locks	The amount of locks per vb hashtable.
ep_ht_size	The initial size of each vb hashtable.
ep_item_num_based_new_chk	True if the number of items in the current checkpoint plays a role in a new checkpoint creation.
ep_keep_closed_chks	True if we want to keep the closed checkpoints for each vbucket unless the memory usage is above high water mark.
ep_max_checkpoints	The maximum amount of checkpoints that can be in memory per vbucket.
ep_max_item_size	The maximum value size.
ep_max_size	The maximum amount of memory this bucket can use.
ep_max_vbuckets	The maximum amount of vbuckets that can exist in this bucket.
ep_mutation_mem_threshold	The ratio of total memory available that we should start sending temp oom or oom message when hitting.
ep_pager_active_vb_pcnt	Active vbuckets paging percentage.
ep_tap_ack_grace_period	The amount of time to wait for a tap acks before disconnecting.
ep_tap_ack_initial_sequence_number	The initial sequence number for a tap ack when a tap stream is created.
ep_tap_ack_interval	The amount of messages a tap producer should send before requesting an ack.
ep_tap_ack_window_size	The maximum amount of ack requests that can be sent before the

	consumer sends a response ack. When the window is full the tap stream is paused..
ep_tap_backfill_resident	The resident ratio for deciding how to do backfill. If under the ratio we schedule full disk backfill. If above the ratio then we do bg fetches for non-resident items.
ep_tap_backlog_limit	The maximum amount of backfill items that can be in memory waiting to be sent to the tap consumer.
ep_tap_backoff_period	The number of seconds the tap connection.
ep_tap_bg_fetch_requeued	Number of times a tap bg fetch task is requeued.
ep_tap_bg_max_pending	The maximum number of bg jobs a tap connection may have.
ep_tap_noop_interval	Number of seconds between a noop is sent on an idle connection.
ep_tap_requeue_sleep_time	The amount of time to wait before a failed tap item is requeued.
ep_tap_throttle_cap_pcnt	Percentage of total items in write queue at which we throttle tap input.
ep_tap_throttle_queue_cap	Max size of a write queue to throttle incoming tap input.
ep_tap_throttle_threshold	Percentage of max mem at which we begin NAKing tap input.
ep_uncommitted_items	The amount of items that have not been written to disk.
ep_vb0	Whether vbucket 0 should be created by default.
ep_waitforwarmup	True if we should wait for the warmup process to complete before enabling traffic.
ep_warmup	Shows if warmup is enabled / disabled.
ep_warmup_batch_size	The size of each batch loaded during warmup.
ep_warmup_dups	Number of Duplicate items encountered during warmup.
ep_warmup_min_items_threshold	Percentage of total items warmed up before we enable traffic.
ep_warmup_min_memory_threshold	Percentage of max mem warmed up before we enable traffic.
ep_warmup_oom	The amount of ‹oom› errors that occurred during warmup.
ep_warmup_thread	The status of the warmup thread.
ep_warmup_time	The amount of time warmup took.

The following table shows replica vbucket statistics.

colspecs...

Stat	Description
vb_replica_num	Number of replica vBuckets.
vb_replica_curr_items	Number of in memory items.
vb_replica_num_non_resident	Number of non-resident items.
vb_replica_perc_mem_resident	% memory resident.
vb_replica_eject	Number of times item values got ejected..
vb_replica_expired	Number of times an item was expired.
vb_replica_ht_memory	Memory overhead of the hashtable.
vb_replica_itm_memory	Total item memory.
vb_replica_meta_data_memory	Total metadata memory.
vb_replica_ops_create	Number of create operations.
vb_replica_ops_update	Number of update operations.
vb_replica_ops_delete	Number of delete operations.
vb_replica_ops_reject	Number of rejected operations.
vb_replica_queue_size	Replica items in disk queue.
vb_replica_queue_memory	Memory used for disk queue.
vb_replica_queue_age	Sum of disk queue item age in milliseconds.
vb_replica_queue_pending	Total bytes of pending writes.
vb_replica_queue_fill	Total enqueued items.
vb_replica_queue_drain	Total drained items.

The following table shows active vbucket statistics:

colspecs...

Stat	Description
vb_active_num	Number of active vBuckets.
vb_active_curr_items	Number of in memory items.
vb_active_num_non_resident	Number of non-resident items.
vb_active_perc_mem_resident	% memory resident.
vb_active_eject	Number of times item values got ejected.
vb_active_expired	Number of times an item was expired.
vb_active_ht_memory	Memory overhead of the hashtable.
vb_active_itm_memory	Total item memory.
vb_active_meta_data_memory	Total metadata memory.
vb_active_ops_create	Number of create operations.
vb_active_ops_update	Number of update operations.
vb_active_ops_delete	Number of delete operations.
vb_active_ops_reject	Number of rejected operations.
vb_active_queue_size	Active items in disk queue.
vb_active_queue_memory	Memory used for disk queue.
vb_active_queue_age	Sum of disk queue item age in milliseconds.
vb_active_queue_pending	Total bytes of pending writes.
vb_active_queue_fill	Total enqueued items.
vb_active_queue_drain	Total drained items.

The following table shows pending vbucket statistics:

colspecs...

Stat	Description
vb_pending_num	Number of pending vBuckets.
vb_pending_curr_items	Number of in memory items.
vb_pending_num_non_resident	Number of non-resident items.
vb_pending_perc_mem_resident	% memory resident.
vb_pending_eject	Number of times item values got ejected.
vb_pending_expired	Number of times an item was expired.
vb_pending_ht_memory	Memory overhead of the hashtable.
vb_pending_itm_memory	Total item memory.
vb_pending_meta_data_memory	Total metadata memory.
vb_pending_ops_create	Number of create operations.
vb_pending_ops_update	Number of update operations.
vb_pending_ops_delete	Number of delete operations.
vb_pending_ops_reject	Number of rejected operations.
vb_pending_queue_size	Pending items in disk queue.
vb_pending_queue_memory	Memory used for disk queue.
vb_pending_queue_age	Sum of disk queue item age in milliseconds.
vb_pending_queue_pending	Total bytes of pending writes.
vb_pending_queue_fill	Total enqueued items.
vb_pending_queue_drain	Total drained items.

Options

The majority of ‹cbstats› commands are predominantly used by Couchbase internally and to help resolve customer support incidents.

The following are the command options:

colspecs...

Options	Description
‹-h, --help›	Shows the help message and exits.
‹-j›	Outputs the result in JSON format
‹-b [BUCKETNAME]›	The bucket to get the status from. Default: ‹default›. If you don't specify a bucket (‹-b xxx›), then you will get information for the ‹default› bucket, if it exists. If the ‹default› bucket does not exist, the output will be empty.
‹-p [PASSWORD]›	The password for the bucket, if one exists.

Examples

When using the Couchbase data port 11210, it will give you operations per node per bucket:

```
$ cbstats -b beer-sample localhost:11210 all | grep \ curr_items:
                            curr_items:                                 3711
```

‹Example with two buckets, ‹default› and ‹c›:›

```
$ cbstats localhost:12000 -b default uuid
--------uuid: 3cd3d2efb20f3b9d7d484925b6d8c03f
--------$ cbstats localhost:12000 uuid
--------uuid: 3cd3d2efb20f3b9d7d484925b6d8c03f
--------$ cbstats localhost:12000 -b c uuid
--------uuid: 1de74b8dc0981abbc6481c5276b91be1
```

Example after the default bucket was deleted, and only the `c` bucket was left:

```
$ cbstats localhost:12000 -b default uuid
--------Authentication error for default
--------$ cbstats localhost:12000 uuid
--------<nothing >
--------$ cbstats localhost:12000 -b c uuid
--------uuid: 1de74b8dc0981abbc6481c5276b91be1
```

Example for getting statistics for timings:

To get statistics, for example, timings on host 10.5.2.117:

```
cbstats 10.5.2.117:11210 timings
```

Response:

```
 disk_commit (1024 total)
    0 - 1s : (100.00%) 1024 ##################################################
    Avg    : (     1s)
 get_stats_cmd (30663276 total)
    0 - 1us       : (  0.05%)    14827
    1us - 2us     : (  6.56%)  1995778 ##
    2us - 4us     : ( 41.79%) 10804626 ##############
    4us - 8us     : ( 45.20%)  1044043 #
    8us - 16us    : ( 45.49%)    89929
    16us - 32us   : ( 45.90%)   124472
    32us - 64us   : ( 46.38%)   148935
    64us - 128us  : ( 56.17%)  2999690 ###
    128us - 256us : ( 68.57%)  3804009 ####
    256us - 512us : ( 69.91%)   411281
    512us - 1ms   : ( 78.77%)  2717402 ###
    1ms - 2ms     : ( 96.36%)  5391526 #######
    2ms - 4ms     : ( 99.05%)   826345 #
    4ms - 8ms     : ( 99.96%)   278727
    8ms - 16ms    : (100.00%)    11443
    16ms - 32ms   : (100.00%)      217
    32ms - 65ms   : (100.00%)       19
    65ms - 131ms  : (100.00%)        7
    Avg           : (  347us)
 disk_vbstate_snapshot (93280 total)
    32us - 64us   : ( 15.34%) 14308 ######
    64us - 128us  : ( 74.74%) 55413 ########################
    128us - 256us : ( 91.39%) 15532 #######
    256us - 512us : ( 95.69%)  4007 #
    512us - 1ms   : ( 99.49%)  3546 #
    1ms - 2ms     : ( 99.95%)   423
    2ms - 4ms     : ( 99.99%)    43
```

```
    4ms - 8ms     : (100.00%)     4
    2s - 4s       : (100.00%)     4
    Avg           : (  190us)
 notify_io (4 total)
    4us - 8us   : ( 25.00%) 1 ############
    16us - 32us : ( 75.00%) 2 ########################
    32us - 64us : (100.00%) 1 ############
    Avg         : (   17us)
```

Example for `cbstats` output:

The `cbstats` output can be used with other command-line tools to sort and filter the data, for example, the `watch` command.

```
watch --diff "cbstats \
    ip-10-12-19-81:11210 -b bucket_name -p bucket_password all | egrep
'item|mem|flusher|ep_queue|bg|eje|resi|warm'"
```

Example for `disk_insert`:

The following sample statistics show that `disk_insert` took 8–16μs 9,488 times, 16–32μs 290 times, and so on.

```
STAT disk_insert_8,16 9488
STAT disk_insert_16,32 290
STAT disk_insert_32,64 73
STAT disk_insert_64,128 86
STAT disk_insert_128,256 48
STAT disk_insert_256,512 2
STAT disk_insert_512,1024 12
STAT disk_insert_1024,2048 1
```

The same statistics displayed by the CLI `cbstats` tool looks like the following:

```
disk_insert (10008 total)
8us - 16us : ( 94.80%) 9488 ##########################################
16us - 32us : ( 97.70%) 290 #
32us - 64us : ( 98.43%) 73
64us - 128us : ( 99.29%) 86
128us - 256us : ( 99.77%) 4
256us - 512us : ( 99.79%) 2
512us - 1ms : ( 99.91%) 12
1ms - 2ms : ( 99.92%) 1
```