Metrics in Prometheus

The integration with Prometheus lets you create dashboards that highlight the metrics that are important to you.

Here are the metrics available to Prometheus:

**Database metrics**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bdb_avg_latency</td>
<td>Average latency of operations on the DB (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_avg_latency_max</td>
<td>Highest value of average latency of operations on the DB (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_avg_read_latency</td>
<td>Average latency of read operations (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_avg_read_latency_max</td>
<td>Highest value of average latency of read operations (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_avg_write_latency</td>
<td>Average latency of write operations (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_avg_write_latency_max</td>
<td>Highest value of average latency of write operations (seconds); returned only when there is traffic</td>
</tr>
<tr>
<td>bdb_conns</td>
<td>Number of client connections to DB</td>
</tr>
<tr>
<td>bdb_egress_bytes</td>
<td>Rate of outgoing network traffic from the DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_egress_bytes_max</td>
<td>Highest value of rate of outgoing network traffic from the DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_evicted_objects</td>
<td>Rate of key evictions from DB (evictions/sec)</td>
</tr>
<tr>
<td>bdb_evicted_objects_max</td>
<td>Highest value of rate of key evictions from DB (evictions/sec)</td>
</tr>
<tr>
<td>bdb_expired_objects</td>
<td>Rate keys expired in DB (expirations/sec)</td>
</tr>
<tr>
<td>bdb_expired_objects_max</td>
<td>Highest value of rate keys expired in DB (expirations/sec)</td>
</tr>
<tr>
<td>bdb_fork_cpu_system</td>
<td>% cores utilization in system mode for all redis shard fork child processes of this database</td>
</tr>
<tr>
<td>bdb_fork_cpu_system_max</td>
<td>Highest value of % cores utilization in system mode for all redis shard fork child processes of this database</td>
</tr>
<tr>
<td>bdb_fork_cpu_user</td>
<td>% cores utilization in user mode for all redis shard fork child processes of this database</td>
</tr>
<tr>
<td>bdb_fork_cpu_user_max</td>
<td>Highest value of % cores utilization in user mode for all redis shard fork child processes of this database</td>
</tr>
<tr>
<td>bdb_ingress_bytes</td>
<td>Rate of incoming network traffic to DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_ingress_bytes_max</td>
<td>Highest value of rate of incoming network traffic to DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_instantaneous_ops_per_sec</td>
<td>Request rate handled by all shards of DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_main_thread_cpu_system</td>
<td>% cores utilization in system mode for all redis shard main threads of this database</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bdb_main_thread_cpu_system_max</td>
<td>Highest value of % cores utilization in system mode for all redis shard main threads of this database</td>
</tr>
<tr>
<td>bdb_main_thread_cpu_user</td>
<td>% cores utilization in user mode for all redis shard main threads of this database</td>
</tr>
<tr>
<td>bdb_main_thread_cpu_user_max</td>
<td>Highest value of % cores utilization in user mode for all redis shard main threads of this database</td>
</tr>
<tr>
<td>bdb_mem_frag_ratio</td>
<td>RAM fragmentation ratio (RSS / allocated RAM)</td>
</tr>
<tr>
<td>bdb_mem_size_lua</td>
<td>Redis lua scripting heap size (bytes)</td>
</tr>
<tr>
<td>bdb_memory_limit</td>
<td>Configured RAM limit for the database</td>
</tr>
<tr>
<td>bdb_monitor_sessions_count</td>
<td>Number of client connected in monitor mode to the DB</td>
</tr>
<tr>
<td>bdb_no_of_keys</td>
<td>Number of keys in DB</td>
</tr>
<tr>
<td>bdb_other_req</td>
<td>Rate of other (non read/write) requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_other_req_max</td>
<td>Highest value of rate of other (non read/write) requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_other_res</td>
<td>Rate of other (non read/write) responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_other_res_max</td>
<td>Highest value of rate of other (non read/write) responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_psub_channels</td>
<td>Count the pub/sub channels with subscribed clients</td>
</tr>
<tr>
<td>bdb_psub_channels_max</td>
<td>Highest value of count the pub/sub channels with subscribed clients</td>
</tr>
<tr>
<td>bdb_psub_patterns</td>
<td>Count the pub/sub patterns with subscribed clients</td>
</tr>
<tr>
<td>bdb_psub_patterns_max</td>
<td>Highest value of count the pub/sub patterns with subscribed clients</td>
</tr>
<tr>
<td>bdb_read_hits</td>
<td>Rate of read operations accessing an existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_hits_max</td>
<td>Highest value of rate of read operations accessing an existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_misses</td>
<td>Rate of read operations accessing a non-existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_misses_max</td>
<td>Highest value of rate of read operations accessing a non-existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_req</td>
<td>Rate of read requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_req_max</td>
<td>Highest value of rate of read requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_res</td>
<td>Rate of read responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_read_res_max</td>
<td>Highest value of rate of read responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_shard_cpu_system</td>
<td>% cores utilization in system mode for all redis shard processes of this database</td>
</tr>
<tr>
<td>bdb_shard_cpu_system_max</td>
<td>Highest value of % cores utilization in system mode for all redis shard processes of this database</td>
</tr>
<tr>
<td>bdb_shard_cpu_user</td>
<td>% cores utilization in user mode for the redis shard process</td>
</tr>
<tr>
<td>bdb_shard_cpu_user_max</td>
<td>Highest value of % cores utilization in user mode for the redis shard process</td>
</tr>
<tr>
<td>bdb_total_connections_received</td>
<td>Rate of new client connections to DB (connections/sec)</td>
</tr>
<tr>
<td>bdb_total_connections_received_max</td>
<td>Highest value of rate of new client connections to DB (connections/sec)</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>bdb_total_req</td>
<td>Rate of all requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_total_req_max</td>
<td>Highest value of rate of all requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_total_res</td>
<td>Rate of all responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_total_res_max</td>
<td>Highest value of rate of all responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_up</td>
<td>Database is up and running</td>
</tr>
<tr>
<td>bdb_used_memory</td>
<td>Memory used by db (in bigredis this includes flash) (bytes)</td>
</tr>
<tr>
<td>bdb_write_hits</td>
<td>Rate of write operations accessing an existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_hits_max</td>
<td>Highest value of rate of write operations accessing an existing</td>
</tr>
<tr>
<td></td>
<td>key (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_misses</td>
<td>Rate of write operations accessing a non-existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_misses_max</td>
<td>Highest value of rate of write operations accessing a non-</td>
</tr>
<tr>
<td></td>
<td>existing key (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_req</td>
<td>Rate of write requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_req_max</td>
<td>Highest value of rate of write requests on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_res</td>
<td>Rate of write responses on DB (ops/sec)</td>
</tr>
<tr>
<td>bdb_write_res_max</td>
<td>Highest value of rate of write responses on DB (ops/sec)</td>
</tr>
<tr>
<td>no_of_expires</td>
<td>Current number of volatile keys in the database</td>
</tr>
</tbody>
</table>

**Node metrics**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_available_flash</td>
<td>Available flash in node (bytes)</td>
</tr>
<tr>
<td>node_available_flash_no_overbooking</td>
<td>Available flash in node (bytes), without taking into account overbooking</td>
</tr>
<tr>
<td>node_available_memory</td>
<td>Amount of free memory in node (bytes) that is available for database</td>
</tr>
<tr>
<td></td>
<td>provisioning</td>
</tr>
<tr>
<td>node_available_memory_no_overbooking</td>
<td>Available ram in node (bytes) without taking into account overbooking</td>
</tr>
<tr>
<td>node_avg_latency</td>
<td>Average latency of requests handled by endpoints on node (seconds);</td>
</tr>
<tr>
<td></td>
<td>returned only when there is traffic</td>
</tr>
<tr>
<td>node_bigstore_free</td>
<td>Sum of free space of back-end flash (used by flash DB’s [BigRedis]) on</td>
</tr>
<tr>
<td></td>
<td>all cluster nodes (bytes); returned only when BigRedis is enabled</td>
</tr>
<tr>
<td>node_bigstore_iops</td>
<td>Rate of i/o operations against back-end flash for all shards which</td>
</tr>
<tr>
<td></td>
<td>are part of a flash based DB (BigRedis) in cluster (ops/sec); returned</td>
</tr>
<tr>
<td></td>
<td>only when BigRedis is enabled</td>
</tr>
<tr>
<td>node_bigstore_kv_ops</td>
<td>Rate of value read/write operations against back-end flash for all shards</td>
</tr>
<tr>
<td></td>
<td>which are part of a flash based DB (BigRedis) in cluster (ops/sec); returned</td>
</tr>
<tr>
<td></td>
<td>only when BigRedis is enabled</td>
</tr>
<tr>
<td>node_bigstore_throughput</td>
<td>Throughput i/o operations against back-end flash for all shards which</td>
</tr>
<tr>
<td></td>
<td>are part of a flash based DB (BigRedis) in cluster (bytes/sec); returned</td>
</tr>
<tr>
<td></td>
<td>only when BigRedis is enabled</td>
</tr>
<tr>
<td>node_conns</td>
<td>Number of clients connected to endpoints on node</td>
</tr>
<tr>
<td>node_cpu_idle</td>
<td>CPU idle time portion (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>node_cpu_idle_max</td>
<td>Highest value of CPU idle time portion (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_idle_median</td>
<td>Average value of CPU idle time portion (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_idle_min</td>
<td>Lowest value of CPU idle time portion (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_system</td>
<td>CPU time portion spent in kernel (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_system_max</td>
<td>Highest value of CPU time portion spent in kernel (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_system_median</td>
<td>Average value of CPU time portion spent in kernel (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_system_min</td>
<td>Lowest value of CPU time portion spent in kernel (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_user</td>
<td>CPU time portion spent by users-pace processes (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_user_max</td>
<td>Highest value of CPU time portion spent by users-pace processes (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_user_median</td>
<td>Average value of CPU time portion spent by users-pace processes (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cpu_user_min</td>
<td>Lowest value of CPU time portion spent by users-pace processes (0-1, multiply by 100 to get percent)</td>
</tr>
<tr>
<td>node_cur_aof_rewrites</td>
<td>Number of aof rewrites that are currently performed by shards on this node</td>
</tr>
<tr>
<td>node_egress_bytes</td>
<td>Rate of outgoing network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_egress_bytes_max</td>
<td>Highest value of rate of outgoing network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_egress_bytes_median</td>
<td>Average value of rate of outgoing network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_egress_bytes_min</td>
<td>Lowest value of rate of outgoing network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_ephemeral_storage_avail</td>
<td>Disk space available to RLEC processes on configured ephemeral disk (bytes)</td>
</tr>
<tr>
<td>node_ephemeral_storage_free</td>
<td>Free disk space on configured ephemeral disk (bytes)</td>
</tr>
<tr>
<td>node_free_memory</td>
<td>Free memory in node (bytes)</td>
</tr>
<tr>
<td>node_ingress_bytes</td>
<td>Rate of incoming network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_ingress_bytes_max</td>
<td>Highest value of rate of incoming network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_ingress_bytes_median</td>
<td>Average value of rate of incoming network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_ingress_bytes_min</td>
<td>Lowest value of rate of incoming network traffic to node (bytes/sec)</td>
</tr>
<tr>
<td>node_persistent_storage_avail</td>
<td>Disk space available to RLEC processes on configured persistent disk (bytes)</td>
</tr>
<tr>
<td>node_persistent_storage_free</td>
<td>Free disk space on configured persistent disk (bytes)</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>node_provisional_flash</td>
<td>Amount of flash available for new shards on this node, taking into account</td>
</tr>
<tr>
<td></td>
<td>overbooking, max redis servers, reserved flash and provision and migration</td>
</tr>
<tr>
<td></td>
<td>thresholds (bytes)</td>
</tr>
<tr>
<td>node_provisional_flash_no_overbooking</td>
<td>Amount of flash available for new shards on this node, without taking into</td>
</tr>
<tr>
<td></td>
<td>account overbooking, max redis servers, reserved flash and provision and</td>
</tr>
<tr>
<td></td>
<td>migration thresholds (bytes)</td>
</tr>
<tr>
<td>node_provisional_memory</td>
<td>Amount of RAM that is available for provisioning to databases out of the</td>
</tr>
<tr>
<td></td>
<td>total RAM allocated for databases</td>
</tr>
<tr>
<td>node_provisional_memory_no_overbooking</td>
<td>Amount of RAM that is available for provisioning to databases, without</td>
</tr>
<tr>
<td></td>
<td>taking into account overbooking</td>
</tr>
<tr>
<td>node_total_req</td>
<td>Request rate handled by endpoints on node (ops/sec)</td>
</tr>
<tr>
<td>node_up</td>
<td>Node is part of the cluster and is connected</td>
</tr>
</tbody>
</table>

**Proxy metrics**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listener_acc_latency</td>
<td>Accumulative latency (sum of the latencies) of all types of commands on</td>
</tr>
<tr>
<td></td>
<td>DB. For the average latency, divide this value by listener_total_res</td>
</tr>
<tr>
<td>listener_acc_latency_max</td>
<td>Highest value of accumulative latency of all types of commands on DB</td>
</tr>
<tr>
<td>listener_acc_other_latency</td>
<td>Accumulative latency (sum of the latencies) of commands that are type</td>
</tr>
<tr>
<td></td>
<td>&quot;other&quot; on DB. For the average latency, divide this value by</td>
</tr>
<tr>
<td></td>
<td>listener_other_res</td>
</tr>
<tr>
<td>listener_acc_other_latency_max</td>
<td>Highest value of accumulative latency of commands that are type &quot;other&quot; on</td>
</tr>
<tr>
<td></td>
<td>DB</td>
</tr>
<tr>
<td>listener_acc_read_latency</td>
<td>Accumulative latency (sum of the latencies) of commands that are type</td>
</tr>
<tr>
<td></td>
<td>&quot;read&quot; on DB. For the average latency, divide this value by</td>
</tr>
<tr>
<td></td>
<td>listener_read_res</td>
</tr>
<tr>
<td>listener_acc_read_latency_max</td>
<td>Highest value of accumulative latency of commands that are type &quot;read&quot; on</td>
</tr>
<tr>
<td></td>
<td>DB</td>
</tr>
<tr>
<td>listener_acc_write_latency</td>
<td>Accumulative latency (sum of the latencies) of commands that are type</td>
</tr>
<tr>
<td></td>
<td>&quot;write&quot; on DB. For the average latency, divide this value by</td>
</tr>
<tr>
<td></td>
<td>listener_write_res</td>
</tr>
<tr>
<td>listener_acc_write_latency_max</td>
<td>Highest value of accumulative latency of commands that are type &quot;write&quot; on</td>
</tr>
<tr>
<td></td>
<td>DB</td>
</tr>
<tr>
<td>listener_auth_cmds</td>
<td>Number of memcached AUTH commands sent to the DB</td>
</tr>
<tr>
<td>listener_auth_cmds_max</td>
<td>Highest value of number of memcached AUTH commands sent to the DB</td>
</tr>
<tr>
<td>listener_auth_errors</td>
<td>Number of error responses to memcached AUTH commands</td>
</tr>
<tr>
<td>listener_auth_errors_max</td>
<td>Highest value of number of error responses to memcached AUTH commands</td>
</tr>
<tr>
<td>listener_cmd_flush</td>
<td>Number of memcached FLUSH_ALL commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_flush_max</td>
<td>Highest value of number of memcached FLUSH_ALL commands sent to the DB</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>listener_cmd_get</td>
<td>Number of memcached GET commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_get_max</td>
<td>Highest value of number of memcached GET commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_set</td>
<td>Number of memcached SET commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_set_max</td>
<td>Highest value of number of memcached SET commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_touch</td>
<td>Number of memcached TOUCH commands sent to the DB</td>
</tr>
<tr>
<td>listener_cmd_touch_max</td>
<td>Highest value of number of memcached TOUCH commands sent to the DB</td>
</tr>
<tr>
<td>listener_conns</td>
<td>Number of clients connected to the endpoint</td>
</tr>
<tr>
<td>listener_egress_bytes</td>
<td>Rate of outgoing network traffic to the endpoint (bytes/sec)</td>
</tr>
<tr>
<td>listener_egress_bytes_max</td>
<td>Highest value of rate of outgoing network traffic to the endpoint (bytes/sec)</td>
</tr>
<tr>
<td>listener_ingress_bytes</td>
<td>Rate of incoming network traffic to the endpoint (bytes/sec)</td>
</tr>
<tr>
<td>listener_ingress_bytes_max</td>
<td>Highest value of rate of incoming network traffic to the endpoint (bytes/sec)</td>
</tr>
<tr>
<td>listener_last_req_time</td>
<td>Time of last command sent to the DB</td>
</tr>
<tr>
<td>listener_last_res_time</td>
<td>Time of last response sent from the DB</td>
</tr>
<tr>
<td>listener_max_connections_exceeded</td>
<td>Number of times the Number of clients connected to the db at the same time has exeeded the max limit</td>
</tr>
<tr>
<td>listener_max_connections_exceeded_max</td>
<td>Highest value of number of times the Number of clients connected to the db at the same time has exeeded the max limit</td>
</tr>
<tr>
<td>listener_monitor_sessions_count</td>
<td>Number of client connected in monitor mode to the endpoint</td>
</tr>
<tr>
<td>listener_other_req</td>
<td>Rate of other (non read/write) requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_other_req_max</td>
<td>Highest value of rate of other (non read/write) requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_other_res</td>
<td>Rate of other (non read/write) responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_other_res_max</td>
<td>Highest value of rate of other (non read/write) responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_other_started_res</td>
<td>Number of responses sent from the DB of type &quot;other&quot;</td>
</tr>
<tr>
<td>listener_other_started_res_max</td>
<td>Highest value of number of responses sent from the DB of type &quot;other&quot;</td>
</tr>
<tr>
<td>listener_read_req</td>
<td>Rate of read requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_read_req_max</td>
<td>Highest value of rate of read requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_read_res</td>
<td>Rate of read responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_read_res_max</td>
<td>Highest value of rate of read responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_read_started_res</td>
<td>Number of responses sent from the DB of type &quot;read&quot;</td>
</tr>
<tr>
<td>listener_read_started_res_max</td>
<td>Highest value of number of responses sent from the DB of type &quot;read&quot;</td>
</tr>
<tr>
<td>listener_total_connections_received</td>
<td>Rate of new client connections to the endpoint (connections/sec)</td>
</tr>
</tbody>
</table>
### Metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listener_total_connections_received_max</td>
<td>Highest value of rate of new client connections to the endpoint (connections/sec)</td>
</tr>
<tr>
<td>listener_total_req</td>
<td>Request rate handled by the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_total_req_max</td>
<td>Highest value of rate of all requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_total_res</td>
<td>Rate of all responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_total_res_max</td>
<td>Highest value of rate of all responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_total_started_res</td>
<td>Number of responses sent from the DB of all types</td>
</tr>
<tr>
<td>listener_total_started_res_max</td>
<td>Highest value of number of responses sent from the DB of all types</td>
</tr>
<tr>
<td>listener_write_req</td>
<td>Rate of write requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_write_req_max</td>
<td>Highest value of rate of write requests on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_write_res</td>
<td>Rate of write responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_write_res_max</td>
<td>Highest value of rate of write responses on the endpoint (ops/sec)</td>
</tr>
<tr>
<td>listener_write_started_res</td>
<td>Number of responses sent from the DB of type &quot;write&quot;</td>
</tr>
<tr>
<td>listener_write_started_res_max</td>
<td>Highest value of number of responses sent from the DB of type &quot;write&quot;</td>
</tr>
</tbody>
</table>

#### Replication metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bdb_replicaof_syncer_ingress_bytes</td>
<td>Rate of compressed incoming network traffic to Replica Of DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_replicaof_syncer_ingress_bytes_decompressed</td>
<td>Rate of decompressed incoming network traffic to Replica Of DB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_replicaof_syncer_local_ingress_lag_time</td>
<td>Lag time between the source and the destination for Replica Of traffic (ms)</td>
</tr>
<tr>
<td>bdb_replicaof_syncer_status</td>
<td>Syncer status for Replica Of traffic; 0 = in-sync, 1 = syncing, 2 = out of sync</td>
</tr>
<tr>
<td>bdb_crdt_syncer_ingress_bytes</td>
<td>Rate of compressed incoming network traffic to CRDB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_crdt_syncer_ingress_bytes_decompressed</td>
<td>Rate of decompressed incoming network traffic to CRDB (bytes/sec)</td>
</tr>
<tr>
<td>bdb_crdt_syncer_local_ingress_lag_time</td>
<td>Lag time between the source and the destination (ms) for CRDB traffic</td>
</tr>
<tr>
<td>bdb_crdt_syncer_status</td>
<td>Syncer status for CRDB traffic; 0 = in-sync, 1 = syncing, 2 = out of sync</td>
</tr>
</tbody>
</table>

#### Shard metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>redis_active_defrag_running</td>
<td>Automatic memory defragmentation current aggressiveness (% cpu)</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>redis_allocator_active</td>
<td>Total used memory including external fragmentation</td>
</tr>
<tr>
<td>redis_allocator_allocated</td>
<td>Total allocated memory</td>
</tr>
<tr>
<td>redis_allocator_resident</td>
<td>Total resident memory (RSS)</td>
</tr>
<tr>
<td>redis_aof_last_cow_size</td>
<td>Last AOF, CopyOnWrite memory</td>
</tr>
<tr>
<td>redis_aof_rewrite_in_progress</td>
<td>The number of simultaneous AOF rewrites that are in progress</td>
</tr>
<tr>
<td>redis_aof_rewrites</td>
<td>Number of AOF rewrites this process executed</td>
</tr>
<tr>
<td>redis_aof_delayed_fsync</td>
<td>Number of times an AOF fsync caused delays in the redis main thread</td>
</tr>
<tr>
<td>redis_blocked_clients</td>
<td>Count the clients waiting on a blocking call</td>
</tr>
<tr>
<td>redis_connected_clients</td>
<td>Number of client connections to the specific shard</td>
</tr>
<tr>
<td>redis_connected_slaves</td>
<td>Number of connected slaves</td>
</tr>
<tr>
<td>redis_db0_avg_ttl</td>
<td>Average TTL of all volatile keys</td>
</tr>
<tr>
<td>redis_db0_expires</td>
<td>Total count of volatile keys</td>
</tr>
<tr>
<td>redis_db0_keys</td>
<td>Total key count</td>
</tr>
<tr>
<td>redis_evicted_keys</td>
<td>Keys evicted so far (since restart)</td>
</tr>
<tr>
<td>redis_expire_cycle_cpu_milliseconds</td>
<td>The cumulative amount of time spent on active expiry cycles</td>
</tr>
<tr>
<td>redis_expired_keys</td>
<td>Keys expired so far (since restart)</td>
</tr>
<tr>
<td>redis_forwarding_state</td>
<td>Shard forwarding state (on or off)</td>
</tr>
<tr>
<td>redis_keys_trimmed</td>
<td>The number of keys that were trimmed in the current or last resharding process</td>
</tr>
<tr>
<td>redis_keyspace_read_hits</td>
<td>Number of read operations accessing an existing keyspace</td>
</tr>
<tr>
<td>redis_keyspace_read_misses</td>
<td>Number of read operations accessing a non-existing keyspace</td>
</tr>
<tr>
<td>redis_keyspace_write_hits</td>
<td>Number of write operations accessing an existing keyspace</td>
</tr>
<tr>
<td>redis_keyspace_write_misses</td>
<td>Number of write operations accessing a non-existing keyspace</td>
</tr>
<tr>
<td>redis_master_link_status</td>
<td>Indicates if the slave is connected to its master</td>
</tr>
<tr>
<td>redis_master_repl_offset</td>
<td>Number of bytes sent to replicas by the shard; Calculate throughput</td>
</tr>
<tr>
<td>redis_master_sync_in_progress</td>
<td>The master shard is synchronizing (1 true)</td>
</tr>
<tr>
<td>redis_max_process_mem</td>
<td>Current memory limit configured by redis_mgr according to node free memory</td>
</tr>
<tr>
<td>redis_maxmemory</td>
<td>Current memory limit configured by redis_mgr according to db memory limits</td>
</tr>
<tr>
<td>redis_mem_aof_buffer</td>
<td>Current size of AOF buffer</td>
</tr>
<tr>
<td>redis_mem_clients_normal</td>
<td>Current memory used for input and output buffers of non-slave clients</td>
</tr>
<tr>
<td>redis_mem_clients_slaves</td>
<td>Current memory used for input and output buffers of slave clients</td>
</tr>
<tr>
<td>redis_mem_fragmentation_ratio</td>
<td>Memory fragmentation ratio (1.3 means 30% overhead)</td>
</tr>
<tr>
<td>redis_mem_not_counted_for_evict</td>
<td>Portion of used_memory (in bytes) that’s not counted for eviction and OOM error</td>
</tr>
<tr>
<td>redis_mem_replication_backlog</td>
<td>Size of replication backlog</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>redis_module_fork_in_progress</td>
<td>A binary value that indicates if there is an active fork spawned by a module (1) or not (0)</td>
</tr>
<tr>
<td>redis_process_cpu_system_seconds_total</td>
<td>Shard Process system CPU time spent in seconds</td>
</tr>
<tr>
<td>redis_process_cpu_usage_percent</td>
<td>Shard Process cpu usage percentage</td>
</tr>
<tr>
<td>redis_process_cpu_user_seconds_total</td>
<td>Shard user CPU time spent in seconds</td>
</tr>
<tr>
<td>redis_process_main_thread_cpu_system_seconds_total</td>
<td>Shard main thread system CPU time spent in seconds</td>
</tr>
<tr>
<td>redis_process_main_thread_cpu_user_seconds_total</td>
<td>Shard main thread user CPU time spent in seconds</td>
</tr>
<tr>
<td>redis_process_max_fds</td>
<td>Shard Maximum number of open file descriptors</td>
</tr>
<tr>
<td>redis_process_open_fds</td>
<td>Shard Number of open file descriptors</td>
</tr>
<tr>
<td>redis_process_resident_memory_bytes</td>
<td>Shard Resident memory size in bytes</td>
</tr>
<tr>
<td>redis_process_start_time_seconds</td>
<td>Shard Start time of the process since unix epoch in seconds</td>
</tr>
<tr>
<td>redis_process_virtual_memory_bytes</td>
<td>Shard virtual memory in bytes</td>
</tr>
<tr>
<td>redis_rdb_bgsave_in_progress</td>
<td>Indication if bgsave is currently in progress</td>
</tr>
<tr>
<td>redis_rdb_last_cow_size</td>
<td>Last bgsave (or SYNC fork) used CopyOnWrite memory</td>
</tr>
<tr>
<td>redis_rdb_saves</td>
<td>Total count of bgsaves since process was restarted (including slave fullsync and persistence)</td>
</tr>
<tr>
<td>redis_repl_touch_bytes</td>
<td>Number of bytes sent to replicas as TOUCH commands by the shard as a result of a READ command that was processed; Calculate the throughput for a time period by comparing the value at different times</td>
</tr>
<tr>
<td>redis_total_commands_processed</td>
<td>Number of commands processed by the shard; Calculate the number of commands for a time period by comparing the value at different times</td>
</tr>
<tr>
<td>redis_total_connections_received</td>
<td>Number of connections received by the shard; Calculate the number of connections for a time period by comparing the value at different times</td>
</tr>
<tr>
<td>redis_total_net_input_bytes</td>
<td>Number of bytes received by the shard; Calculate the throughput for a time period by comparing the value at different times</td>
</tr>
<tr>
<td>redis_total_net_output_bytes</td>
<td>Number of bytes sent by the shard; Calculate the throughput for a time period by comparing the value at different times</td>
</tr>
<tr>
<td>redis_up</td>
<td>Shard is up and running</td>
</tr>
<tr>
<td>redis_used_memory</td>
<td>Memory used by shard (in bigredis this includes flash) (bytes)</td>
</tr>
</tbody>
</table>

**Updated:** December 6, 2020