|  |  |  |
| --- | --- | --- |
| **Host test S.O** | Ubuntu | |
| **Version** | 18.04.2 LTS | |
| **Quantity of widgets created for default** | 8 | |
| **View general** | | |
|  | | |
| **Widgets** | | |
| **CPU USAGE** | | |
|  | | |
| **Wigdet** | | Timeseries-area |
| **Promql** | | **Metrics to evaluate** |
| * avg:system.cpu.idle{host:ubuntu-VirtualBox} * avg:system.cpu.system{host:ubuntu-VirtualBox} * vg:system.cpu.iowait{host:ubuntu-VirtualBox} * avg:system.cpu.user{host:ubuntu-VirtualBox} * avg:system.cpu.stolen{host:ubuntu-VirtualBox} * avg:system.cpu.guest{host:ubuntu-VirtualBox} | | * Percent of time the CPU spent in an idle state * The percent of time the CPU spent running the kernel * The percent of time the CPU spent waiting for IO operations to complete. * The percent of time the CPU spent running user space processes. * The percent of time the virtual CPU spent waiting for the hypervisor to service another virtual CPU. Only applies to virtual machines * The percent of time the CPU spent running the virtual processor. Only applies to hypervisors |
| **Processes memory usage** | | |
|  | | |
| **Wigdet** | | TreeMap |
| **Promql** | | **Metrics to evaluate** |
| * processes{host:ubuntu-VirtualBox} | | * This widget evaluates metric of process memory usage |
| **Load averages 1-5-15** | | |
|  | | |
| **Widget** | | Timeseries-lines |
| **Promql** | | **Metrics to evaluate** |
| * system.load.1{host:ubuntu-VirtualBox} * system.load.5{host:ubuntu-VirtualBox} * system.load.15{host:ubuntu-VirtualBox} | | * The average system load over one minute * The average system load over five minutes. * The average system load over fifteen minutes. |
| **Memory breakdown** | | |
|  | | |
| **Widget** | | Timeseries-area |
| **Promql** | | **Metrics to evaluate** |
| * sum:system.mem.usable{host:ubuntu-VirtualBox} * sum:system.mem.total{host:ubuntu-VirtualBox} * sum:system.mem.total{host:ubuntu-VirtualBox} - sum:system.mem.usable{host:ubuntu-VirtualBox} | | * Value of MemAvailable from /proc/meminfo if present, but falls back to adding free + buffered + cached memory if not. * The total amount of physical RAM * This query involucre one operation of sustraction between metrics |
| **Available Swap** | | |
|  | | |
| **Wigdet** | | Timeseries-area |
| **Promql** | | **Metrics to evaluate** |
| * avg:system.swap.free{host:ubuntu-VirtualBox} * avg:system.swap.used{host:ubuntu-VirtualBox} | | * The amount of free swap space. * The amount of swap space in use. |
| **Disk usage by device %** | | |
|  | | |
| **Wigdet** | | Timeseries-lines |
| **Promql** | | **Metrics to evaluate** |
| * max:system.disk.in\_use{host:ubuntu-VirtualBox}by{device} \* 100 | | * The amount of disk space in use as a fraction of the total. |
| **Disk latency (by device)** | | |
|  | | |
| **Widget** | | Timeseries-area |
| **Promql** | | **Metrics to evaluate** |
| * max:system.io.await{host:ubuntu-VirtualBox}by{device} | | * The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them (Linux only). |
| **Network traffic (bytes per sec)** | | |
|  | | |
| **Widget** | | timeseries |
| **Promql** | | **Metrics to evaluate** |
| * sum:system.net.bytes\_rcvd{host:ubuntu-VirtualBox} * 0 - sum:system.net.bytes\_sent{host:ubuntu-VirtualBox} | | * The number of bytes received on a device per second. * The number of bytes sent from a device per second. |

**Integrations**

|  |  |  |
| --- | --- | --- |
| **Integration** | | **MYSQL** |
| **Version** | | **14.4** |
| **Quantity of widgets created for default** | | **11** |
| **View general** | | |
|  | | |
| **Widgets** | | |
| **MYSQL connections** | | |
|  | | |
| **Wigdet** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:mysql.net.connections{$scope} * sum:mysql.net.max\_connections{$scope} | * The rate of connections to the server. * The maximum number of connections that have been in use simultaneously since the server started. | |
| **MYSQL reads and writes (per sec)** | | |
|  | | |
| Wigdet | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:mysql.innodb.data\_reads{$scope} * sum:mysql.innodb.data\_writes{$scope} | * The rate of data reads. * The rate of data writes. | |
| **MySQL fsync op count (per sec)** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:mysql.innodb.os\_log\_fsyncs{$scope} | * The rate of fsync writes to the log file | |
| **MySQL slow queries** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:mysql.performance.slow\_queries{$scope} | * The rate of slow queries. | |
| **MySQL locking rate (per sec)** | | |
|  | | |
| **Wigdet** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * rate(sum:mysql.performance.table\_locks\_waited{$scope}) | * The total number of times that a request for a table lock could not be granted immediately and a wait was needed | |
| **MySQL CPU time (per sec)** | | |
|  | | |
| **Wigdet** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * avg:mysql.performance.user\_time{$scope} | * Percentage of CPU time spent in user space by MySQL. | |
| **CPU USAGE** | | |
|  | | |
| **Widget** | Timeseries-area | |
| **Promql** | **Metrics to evaluate** | |
| * avg:system.cpu.idle{$scope} * avg:system.cpu.system{$scope} * vg:system.cpu.iowait{$scope} * avg:system.cpu.user{$scope} * avg:system.cpu.stolen{$scope} * avg:system.cpu.guest{$scope} | * Percent of time the CPU spent in an idle state * The percent of time the CPU spent running the kernel * The percent of time the CPU spent waiting for IO operations to complete. * The percent of time the CPU spent running user space processes. * The percent of time the virtual CPU spent waiting for the hypervisor to service another virtual CPU. Only applies to virtual machines * The percent of time the CPU spent running the virtual processor. Only applies to hypervisors | |
| **I / O wait** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * max:system.cpu.iowait{$scope} | * The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them (Linux only). | |
| **System load** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * avg:system.load.1{$scope} * avg:system.load.5{$scope} * avg:system.load.15{$scope} | * The average system load over one minute * The average system load over five minutes. * The average system load over fifteen minutes. | |
| **System memory** | | |
|  | | |
| **Widget** | Timeseries-area | |
| **Promql** | **Metrics to evaluate** | |
| * sum:system.mem.usable{$scope} * sum:system.mem.total{$scope} * sum:system.mem.total{$scope}- sum:system.mem.usable{$scope} | * Value of MemAvailable from /proc/meminfo if present but falls back to adding free + buffered + cached memory if not. * The total amount of physical RAM * this query involucre one operation of subtraction between metrics | |
| **Network traffic (bytes per sec)** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:system.net.bytes\_rcvd{$scope} * sum:system.net.bytes\_sent{$scope} | * The number of bytes received on a device per second. * The number of bytes sent from a device per second. | |

|  |  |  |
| --- | --- | --- |
| **Integration** | | Redis |
| **Version** | | 4.0.9 |
| **Quantity of widgets created for default** | | 28 |
| **View general** | | |
|  | | |
| **Widgets** | | |
| **Hit rate** | | |
|  | | |
| **Widget** | Query value | |
| **Promql** | **Metrics to evaluate** | |
| * avg:redis.stats.keyspace\_hits{$scope} * avg:redis.stats.keyspace\_misses{$scope} | * The rate of successful lookups in the main db. * The rate of missed lookups in the main db.   Additional this query included one operation that definy how:  **( a / ( a + b ) ) \* 100**  Where a is keyspace\_hits and b is keyspace\_misses | |
| **Blocked clients** | | |
|  | | |
| **Widget** | Query value | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.clients.blocked{$scope} | * The number of connections waiting on a blocking call | |
| **Redis keyspace** | | |
|  | | |
| **Wigdet** | Query value | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.keys{$scope} | * The total number of keys. | |
| **Unsaved changes** | | |
|  | | |
| **Wigdet** | Query value | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.rdb.changes\_since\_last{$scope} | * The number of changes since the last background save. | |
| **Master link down** | | |
|  | | |
| **Widget** | Query value | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.replication.master\_link\_down\_since\_seconds{$scope} | * Amount of time that the master link has been down. | |
| **Commands per second** | | |
|  | | |
| Wigdet | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.net.commands{$scope} | * The number of commands processed by the server | |
| Evictions | | |
|  | | |
| **Wigdet** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * per\_minute(sum:redis.keys.evicted{$scope}) | * The total number of keys evicted due to the maxmemory limit | |
| **Connected slaves** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.net.slaves{$scope} | * The number of connected slaves | |
| **Cache hit rate** | | |
|  | | |
| **Widget** | timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * avg:redis.stats.keyspace\_hits{$scope} * avg:redis.stats.keyspace\_misses{$scope} | * The rate of successful lookups in the main db. * The rate of missed lookups in the main db.   Additional include operation of define how: **( a / ( a + b ) ) \* 100** where a is keyspace\_hits and b is keyspace\_misses | |
| **Fragmentation ratio** | | |
|  | | |
| **Widget** | timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * avg:redis.mem.fragmentation\_ratio{$scope} | * Ratio between used\_memory\_rss and used\_memory | |
| **Rejected connections** | | |
|  | | |
| Widget | timeseries-lines | |
| Promql | * diff(sum:redis.net.rejected{$scope}) | |
| Metrics to evaluate | * The number of rejected connections | |
| **Slowlog** | | |
|  | | |
| **Widget** | timeseries-area | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.slowlog.micros.95percentile{$scope} by {name,command} | * The 95th percentile of the duration of queries reported in the slow log.   Additional this widget have the next operation: top( a / 1000 , 10 , 'mean' , 'desc' ) | |
| **Blocked clients** | | |
|  | | |
| **Widget** | timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * avg:redis.clients.blocked{$scope} | * The number of connections waiting on a blocking call. | |
| **Latency** | | |
|  | | |
| **Widget** | Timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| avg:redis.info.latency\_ms{$scope} | * The latency of the redis INFO command | |
| **Used memory** | | |
|  | | |
| **Widget** | timeseries-area | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.mem.used{$scope} | * Amount of memory allocated by Redis | |
| **Connected clients** | | |
|  | | |
| **Widget** | timeseries-lines | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.net.clients{$scope} | * The number of connected clients (excluding slaves) | |
| **Keys** | | |
|  | | |
| **Widget** | * timeseries-area | |
| **Promql** | **Metrics to evaluate** | |
| * sum:redis.keys{$scope} | * The total number of keys | |
| **Redis logs** | | |
|  | | |
| **Widget** | Log stream | |
| **Promql** | **Metrics to evaluate** | |
| Not applicable | The Log Stream displays a log flow matching the defined query | |
| Additional dashboards redis contains free text that have objective explain some widgets,display logo and functioning in general | | |