Aerospike Metrics Organization

1. Namespace Metrics

Namespace-related metrics typically deal with the health and behavior of specific namespaces in Aerospike.

General Health:

- `namespace_objects` – Total number of objects in the namespace.
- `free-pct-disk` – Percentage of free disk space.
- `free-pct-memory` – Percentage of free memory.
- `stop-writes` – Indicator if writes are stopped due to low resources.

Replication and Consistency:

- `master_objects` – Number of master objects.
- `replica_objects` – Number of replica objects.
- `non_replica_objects` – Number of non-replicated objects.
- `unavailable_partitions` – Partitions currently unavailable.

Performance and Latency:

- `client_read_success` – Number of successful client reads.
- `client_write_success` – Number of successful client writes.
- `read-time-ms` – Latency for reads (ms).
- `write-time-ms` – Latency for writes (ms).

Storage and Eviction:

- `device_used_bytes` – Bytes used on storage devices.
- `device_available_pct` – Percentage of device storage available.
- `evicted_objects` – Total number of evicted objects.

---

2. Node Metrics

Node-level metrics reflect the overall system health and operational efficiency of individual nodes.

Health and Resource Utilization:

- `uptime` – Time since the node last restarted.
- `free-disk-bytes` – Free disk space in bytes.
- `free-mem-bytes` – Free memory in bytes.
- `cluster_size` – Number of nodes in the cluster.

Replication and Partition Metrics:

- `partition_count` – Total number of partitions handled by the node.
- `prole_objects` – Objects the node holds as a replica.

Operations and Throughput:

- `batch_sub_read_success` – Number of successful batch reads.
- `read_success` – Total successful reads.
- `write_success` – Total successful writes.
- `query_reqs` – Number of query requests.

---

3. Performance Metrics

Metrics related to latency, throughput, and efficiency in handling requests.

Latency:

- `query_latency` – Latency for query operations.
- `read-time-ms` – Time spent processing read operations (ms).
- `write-time-ms` – Time spent processing write operations (ms).

Client Operations:

- `client_read_error` – Number of read errors from clients.
- `client_write_error` – Number of write errors from clients.
- `batch_sub_tps` – Transactions per second for batch requests.

Disk and Network Throughput:

- `device_write_bytes` – Bytes written to storage.
- `device_read_bytes` – Bytes read from storage.
- `network_send_kbps` – Network traffic sent in KB per second.
- `network_receive_kbps` – Network traffic received in KB per second.

---

4. Storage and Device Metrics

Focuses on disk utilization and device efficiency.

Disk Usage:

- `used-bytes-disk` – Disk space used.
- `free-bytes-disk` – Free disk space available.

- `hwm-disk-pct` – High watermark for disk usage.

Device Operations:

- `device_write_q` – Queue length for writes.
- `device_read_q` – Queue length for reads.
- `defrag_q` – Defragmentation queue length.
- `defrag_reads` – Reads performed for defragmentation.
- `defrag_writes` – Writes performed for defragmentation.

---

5. Query Metrics

Metrics that focus on query performance and reliability.

Performance:

- `query_reqs` – Number of queries initiated.
- `query_fail` – Number of failed queries.
- `query_abort` – Number of queries aborted.

Execution Metrics:

- `query_rec_count` – Number of records processed during queries.
- `query_replica` – Indicates whether the query uses replicas.

---

6. Security Metrics

Tracks user authentication and system-level security metrics.

Authentication:

- `auth_success` – Number of successful authentications.
- `auth_fail` – Number of failed authentications.

Access Control:

- `acl_reads_denied` – Reads denied by access control.
- `acl_writes_denied` – Writes denied by access control.

---

7. Miscellaneous Metrics

Miscellaneous or less commonly used metrics.

- `strong_consistency_failures` – Number of consistency-related failures.

- `scan_basic_complete` – Number of completed basic scans.
- `scan_basic_abort` – Number of aborted scans.
- `heartbeat_connections` – Active heartbeat connections between nodes.
- `xdr_write_reqs` – Number of XDR write requests.

Metrics Testable with `asbench`:

1. Performance Metrics

- **Throughput:**
  - `read_success` and `write_success`: These can be evaluated by observing the number of successful read and write operations during a benchmark test.
  - `batch_sub_tps`: Measures transactions per second for batch requests, which `asbench` can help simulate.
- **Latency:**
  - `read-time-ms` and `write-time-ms`: `asbench` provides insights into the average latency of read and write operations.

2. Client Operations

- **Error Metrics:**
  - `client_read_error` and `client_write_error`: Can be observed under high load or during stress testing when `asbench` generates intense client activity.

3. Node Metrics

- **Throughput and Resource Usage:**
  - `read_success` and `write_success`: Measure node-level throughput under the load generated by `asbench`.
  - `batch_sub_read_success`: Helps analyze batch operation success rates if the benchmark includes batch reads.

4. Query Metrics

- If the benchmark includes query simulation (if supported by your version of `asbench`):
  - `query_reqs`: Number of queries initiated.
  - `query_fail`: Number of failed queries due to excessive load.

5. Namespace Metrics

- **Replication and Consistency:**
  - Metrics such as `master_objects`, `replica_objects`, and `non_replica_objects` can indirectly be affected by the workload generated using `asbench`, especially when the benchmark affects replication consistency under heavy writes.

6. Storage Metrics

While `asbench` is not specifically tailored to measure storage metrics, heavy workloads might provide indirect insights into:

- `device_write_bytes` and `device_read_bytes`: Bytes written/read to storage during the benchmark test.

How to Use `asbench` for Testing These Metrics

- Use `asbench` to run a workload (e.g., a mix of reads and writes) and collect metrics during the test using the Aerospike Monitoring tools (`aerospike.log`, `asadm`, etc.).
- Analyze system-level and namespace metrics via `asadm` or `aerospike.log` during and after the test to correlate with the benchmark's generated load.

Metrics Testable with `asadm`:

1. Namespace Metrics

`asadm` can fetch detailed namespace-level information using the `show namespace` or `info namespace` commands.

- **General Health:**
  - `namespace_objects`: Total number of objects in the namespace.
  - `free-pct-disk` and `free-pct-memory`: Disk and memory availability percentages.
  - `stop-writes`: Indicates if writes are stopped due to low resources.
- **Replication and Consistency:**
  - `master_objects`, `replica_objects`, and `non_replica_objects`: Provide information on data distribution and replication.
  - `unavailable_partitions`: Number of partitions unavailable, indicating consistency issues.
- **Performance and Latency:**
  - Metrics like `client_read_success`, `client_write_success`, `read-time-ms`, and `write-time-ms` can be monitored to evaluate namespace-level performance.
- **Storage and Eviction:**
  - `device_used_bytes` and `device_available_pct`: Storage consumption and availability.
  - `evicted_objects`: Track object evictions to understand memory pressure.

---

2. Node Metrics

Using `asadm` commands like `show config` or `show stat`, you can gather node-specific metrics.

- **Health and Resource Utilization:**
  - `uptime`: Time since the node was last restarted.
  - `free-disk-bytes` and `free-mem-bytes`: Remaining disk and memory resources.
  - `cluster_size`: Current size of the Aerospike cluster.
- **Replication and Partition Metrics:**
  - `partition_count`: Total number of partitions handled by the node.
  - `prole_objects`: Number of replica objects the node is responsible for.
- **Operations and Throughput:**
  - `batch_sub_read_success`, `read_success`, and `write_success`: Total operations completed.
  - `query_reqs`: Number of query requests served.

---

3. Performance Metrics

Performance metrics are vital for assessing system throughput and response times. Use commands like `show stat` to retrieve these.

- **Latency:**
  - `read-time-ms` and `write-time-ms`: Observe latency for read/write operations.
  - `query_latency`: Monitor query operation latencies.
- **Client Operations:**
  - `client_read_error` and `client_write_error`: Identify errors in client read/write operations.
- **Disk and Network Throughput:**
  - `device_write_bytes` and `device_read_bytes`: Measure I/O activity on storage devices.
  - `network_send_kbps` and `network_receive_kbps`: Track network throughput.

---

4. Storage and Device Metrics

The `show stat` command also provides device-level details.

- **Disk Usage:**
  - `used-bytes-disk` and `free-bytes-disk`: Monitor storage usage and free space.
  - `hwm-disk-pct`: Indicates the high-watermark threshold for disk usage.
- **Device Operations:**
  - `device_write_q` and `device_read_q`: Queue lengths for storage operations.

- `defrag_q`, `defrag_reads`, and `defrag_writes`: Defragmentation statistics.

---

5. Query Metrics

Query metrics can be fetched using `asadm` commands focused on query statistics.

- **Performance:**
  - `query_reqs`: Total number of query requests.
  - `query_fail` and `query_abort`: Identify failed or aborted queries.
- **Execution Metrics:**
  - `query_rec_count`: Number of records processed during queries.
  - `query_replica`: Monitor whether queries use replica data.

---

6. Security Metrics

Use `asadm` commands like `info security` to fetch security-related metrics.

- **Authentication:**
  - `auth_success` and `auth_fail`: Count of successful and failed authentications.
- **Access Control:**
  - `acl_reads_denied` and `acl_writes_denied`: Number of access control rejections.

---

7. Miscellaneous Metrics

`asadm` can also retrieve various other metrics for deeper insights.

- `strong_consistency_failures`: Failures related to strong consistency.
- `scan_basic_complete` and `scan_basic_abort`: Track scan operations.
- `heartbeat_connections`: Number of active heartbeat connections in the cluster.
- `xdr_write_reqs`: Number of XDR write requests for cross-datacenter replication.

Using `asadm` Effectively

To target these metrics:

- Use `asadm` commands such as `show stat`, `info`, `show namespace`, `show config`, and `show distribution`.
- For live monitoring, use `watch` mode in `asadm` to track real-time metric changes.
- Combine `asadm` outputs with tools like `grep` or custom scripts to filter and focus on specific metrics.

If you need guidance on specific `asadm` commands or examples for gathering these metrics, let me know!