Scylla 1.7, Scylla 2.0 Monitoring CheatSheet

For simplicity, the following commands assume that your binaries are in a directory in your $PATH

SCYLLA.

Nodetool Commands (more info here)

General Monitoring Commands	
Metric description	**Command**
High-level status of the cluster	nodetool status
Stats from a specific node	nodetool --host <ip_address> info / nodetool info (runs locally)
Stats on all keyspaces and column families	nodetool cfstats / tablestats
Stats from a specific keyspace	nodetool cfstats / tablestats <keyspace_name>
Stats from a specific column family / table	nodetool cfstats / tablestats <keyspace_name>**.**<column_family / table_name>
Latency stats from specific table	nodetool cfhistograms / tablehistograms <keyspace_name> <column_family / table_name>
Gossip info for the cluster (schema ID)	nodetool --host <ip_address> gossipinfo / nodetool gossipinfo (runs locally)
Compaction Metrics	
Metric description	**Command**
Compactions in progress, pending compaction tasks	nodetool compactionstats

Useful Linux Commands (more info here)

Display disk utilization	iostat -x 2
CPU time breakdowns per CPU	mpstat -P ALL 2
Check network interface throughput	sar -n DEV 2
Virtual memory statistics	vmstat 2

* May require `sysstat` package

Scylla Service (more info here)

Filter only Scylla logs	journalctl _COMM=scylla / journalctl _UID=`id -u scylla`
Stop / Start Scylla server	sudo systemctl stop/start scylla-server

Prometheus Metrics (more info here)

Query Scylla Metrics	
Metric Description	**Query**
Average load per server (per sec)	avg(scylla_reactor_utilization{}) by (instance)
Load per shard (CPU) per sec	scylla_reactor_utilization{}
Total requests (per sec)	CQL: sum(irate(scylla_transport_requests_served{}[30s]))
	Thrift: sum(irate(scylla_thrift_served{}[30s]))
Request served per server (per sec)	CQL: sum(irate(scylla_transport_requests_served{}[30s])) by (instance)
	Thrift: sum(irate(scylla_thrift_served{}[30s])) by (instance)
Total reads (per sec)	sum(irate(scylla_database_total_reads{}[30s])) by (instance)
Total writes (per sec)	sum(irate(scylla_database_total_writes{}[30s])) by (instance)
Running compactions	sum(scylla_compaction_manager_compactions{}) by (instance)
Query Disk Metrics	
Metric Description	**Query**
Disk writes Bps per server	irate(node_disk_bytes_written{device="$monitor_disk"}[30s])
Disk reads Bps per server	irate(node_disk_bytes_read{device="$monitor_disk"}[30s])
Total Capacity (bytes)	sum(node_filesystem_avail{mountpoint="/var/lib/scylla"})
Used capacity (bytes)	sum(node_filesystem_size{mountpoint="/var/lib/scylla"})-sum(node_filesystem_avail{mountpoint="/var/lib/scylla"})

* Mountpoint may differ on your installation

JMX Metrics via JConsole / REST-API

Query Scylla Metrics			
Metric Description	**JMX Path**	**Attribute**	**REST-API (Swagger UI: http://[API_address]:10000/ui)**
Reads per second (avg over previous 1 min)	org.apache.cassandra.metrics:type=ClientRequest,scope=Read,name=Latency	OneMinuteRate	**URI:** http://[API_address]:10000/storage_proxy/metrics/read/moving_average_histogram
Writes per second (avg over previous 1 min)	org.apache.cassandra.metrics:type=ClientRequest,scope=Write,name=Latency	OneMinuteRate	**URI:** http://[API_address]:10000/storage_proxy/metrics/write/moving_average_histogram
Reads (total count)	org.apache.cassandra.metrics:type=ClientRequest,scope=Read,name=Latency	Count	**URI:** http://[API_address]:10000/storage_proxy/metrics/read/histogram
Write (total count)	org.apache.cassandra.metrics:type=ClientRequest,scope=Write,name=Latency	Count	**URI:** http://[API_address]:10000/storage_proxy/metrics/write/histogram
Read latency (total elapsed time in µs)	org.apache.cassandra.metrics:type=ClientRequest,scope=Read,name=TotalLatency	Count	**URI:** http://[API_address]:10000/storage_proxy/metrics/read
Write latency (total elapsed time in µs)	org.apache.cassandra.metrics:type=ClientRequest,scope=Write,name=TotalLatency	Count	**URI:** http://[API_address]:10000/storage_proxy/metrics/write
Row cache hits	org.apache.cassandra.metrics:name=Hits,scope=RowCache,type=Cache	Count	**URI:** http://[API_address]:10000/cache_service/metrics/row/hits
Row cache requests	org.apache.cassandra.metrics:name=Requests,scope=RowCache,type=Cache	Count	**URI:** http://[API_address]:10000/cache_service/metrics/row/requests
Row cache hit rate (avg over previous 5 min)	org.apache.cassandra.metrics:name=Hits,scope=RowCache,type=Cache	FiveMinuteRate	**URI:** http://[API_address]:10000/cache_service/metrics/row/hits_moving_avrage

Query Disk Metrics			
Metric Description	**JMX Path**	**Attribute**	**REST-API**
Disk used on a node (bytes)	org.apache.cassandra.metrics:type=ColumnFamily,name=TotalDiskSpaceUsed	Count	**URI:** http://[API_address]:10000/column_family/metrics/total_disk_space_used
Disk used by a column family / table (bytes)	org.apache.cassandra.metrics:type=ColumnFamily,keyspace=<keyspace>,scope=<table>,name=TotalDiskSpaceUsed	Count	**URI:**http://[API_address]:10000/column_family/metrics/total_disk_space_used/<keyspace>%3A<table>
Completed compaction tasks	org.apache.cassandra.metrics:type=Compaction,name=CompletedTasks	Value	**URI:** http://[API_address]:10000/compaction_manager/metrics/completed_tasks
Pending compaction tasks	org.apache.cassandra.metrics:type=Compaction,name=PendingTasks	Value	**URI:** http://[API_address]:10000/compaction_manager/metrics/pending_tasks

* JMX Path: `ColumnFamily` can be replaced with `Table`