

Cassandra Healthcheck Example (Instaclustr)

Professional Services

Table of Contents

1. Operational Health	1
1.1. JVM Heap Usage (Prometheus)	1
1.2. CPU Utilization (Prometheus)	1
1.3. Disk Usage (Prometheus)	1
1.4. Compaction Pending Tasks (Prometheus)	1
1.5. Read/Write Latency P95 (Prometheus)	2
1.6. Table Statistics	2
1.7. Read Repair Settings	2
1.8. Secondary Indexes	2
1.9. Network Topology	2
1.10. Compaction Pending Tasks Analysis (All Nodes)	2
1.11. Schema Version Consistency (All Nodes)	3
1.12. Disk Space per Keyspace and Table	3
1.13. Temporary Files in Data Directory (All Nodes)	5
1.14. Tombstone Metrics Analysis	6
1.15. GC Grace Seconds Audit	6
1.16. Table Compression Settings Analysis	9
1.17. Row Cache Analysis	16
1.18. Durable Writes Analysis	19
1.19. Materialized Views Analysis	20
1.20. User-Defined Functions and Aggregates Analysis	20
1.21. Keyspace Replication Health Analysis	20
1.22. Superuser Roles Analysis	22
2. Configuration	23
2.1. Keyspace Replication Strategy Analysis	23
3. Hardware	25
3.1. AI-Generated Recommendations	25
3.2. AI-Generated Recommendations	25

Chapter 1. Operational Health

Pulling metrics from IC Prometheus Endpoint

1.1. JVM Heap Usage (Prometheus)



□ All 3 nodes have healthy heap usage (<75%)

Cluster Summary:

- Total Heap: 0.86 GB / 1.5 GB
- Cluster Usage: 57.32%
- Nodes Monitored: 3

1.2. CPU Utilization (Prometheus)



□ All 3 nodes have healthy CPU usage (<75%)

Cluster Summary:

- Average CPU: 8.96%
- Max CPU: 9.62%
- Min CPU: 8.62%
- Nodes Monitored: 3

1.3. Disk Usage (Prometheus)



□ All 3 nodes have healthy disk usage (<70%)

Cluster Summary:

- Average Disk: 0.8%
- Max Disk: 0.8%
- Min Disk: 0.8%
- Nodes Monitored: 3

1.4. Compaction Pending Tasks (Prometheus)



□ No pending compactions across 3 nodes

Cluster Summary:

- Total Pending: 0 tasks

- Average Pending: 0.0 tasks/node
- Max Pending: 0 tasks
- Nodes with Pending: 0/3

1.5. Read/Write Latency P95 (Prometheus)



□ Latency healthy across all nodes (p95 read <50ms, write <30ms)

Cluster Summary:

- Avg Read Latency (p95): 5.79 ms
- Avg Write Latency (p95): 4.05 ms
- Max Read Latency: 11.86 ms
- Max Write Latency: 6.87 ms

1.6. Table Statistics



□ Analyzed 16 table(s) across 7 keyspace(s)

1.7. Read Repair Settings



□□ 16 of 16 table(s) have non-recommended read repair settings

1.8. Secondary Indexes



□□ Found 2 secondary index(es): 2 standard, 0 SASI, 0 custom

1.9. Network Topology



□□ 1 topology concern(s): 3 node(s) across 1 datacenter(s) and 2 rack(s)

1.10. Compaction Pending Tasks Analysis (All Nodes)

Checking for pending compaction tasks using `nodetool compactionstats` across all nodes.



Requirements:

- SSH access to the database server



This check requires SSH access for `nodetool` commands.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address

For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

1.11. Schema Version Consistency (All Nodes)

Verifying that all nodes in the cluster agree on the schema version using `nodetool describeclasser`.



Requirements:

- SSH access to the database server

This check requires SSH access for `nodetool` commands.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

1.12. Disk Space per Keyspace and Table

Analyzing disk usage across keyspaces and tables using `nodetool tablestats`. This check aggregates live disk space usage, excluding system keyspaces.



Requirements:

- SSH access to the database server

This check requires SSH access for `nodetool` commands.



Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address

For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for Disk space check.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for Memory usage check.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for JVM statistics check.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for CPU load check.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

1.13. Temporary Files in Data Directory (All Nodes)

Scanning for temporary files in Cassandra data directories using `find` command across all nodes.



Requirements:

- SSH access to the database server

This check requires SSH access for shell commands.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for GC stats check.

Configure the following in your settings:



For single host: * `ssh_host`: Hostname or IP address

For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

1.14. Tombstone Metrics Analysis

Checking for high tombstone counts across tables using `nodetool tablestats`.



Requirements:

- SSH access to the database server

This check requires SSH access for `nodetool` commands.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

This check requires SSH access for Cluster connectivity diagnostics.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address



For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

1.15. GC Grace Seconds Audit

Scanning all tables for inappropriate `gc_grace_seconds` settings (> 3 days or 0).

keyspace_name	table_name	gc_grace_seconds
---------------	------------	------------------

test_indexes	users_indexed	864000
system_auth	cidr_groups	7776000
system_auth	cidr_permissions	7776000
system_auth	identity_to_role	7776000
system_auth	network_permissions	7776000
system_auth	resource_role_permissions_index	7776000
system_auth	role_members	7776000
system_auth	role_permissions	7776000
system_auth	roles	7776000
system_schema	aggregates	604800
system_schema	column_masks	604800
system_schema	columns	604800
system_schema	dropped_columns	604800
system_schema	functions	604800
system_schema	indexes	604800
system_schema	keyspaces	604800
system_schema	tables	604800
system_schema	triggers	604800
system_schema	types	604800
system_schema	views	604800
test_load	load_test_data	864000
test_compaction	lcs_table	864000
test_compaction	stcs_table	864000
test_compaction	twcs_table	864000
system_distributed	parent_repair_history	864000
system_distributed	partition_denylist	864000
system_distributed	repair_history	864000
system_distributed	view_build_status	864000
system	IndexInfo	0
system	available_ranges	0
system	available_ranges_v2	0
system	batches	0
system	built_views	0
system	compaction_history	0

system	local	0
system	paxos	0
system	paxos_repair_history	0
system	peer_events	0
system	peer_events_v2	0
system	peers	0
system	peers_v2	0
system	prepared_statements	0
system	repairs	0
system	size_estimates	0
system	sstable_activity	0
system	sstable_activity_v2	0
system	table_estimates	0
system	top_partitions	0
system	transferred_ranges	0
system	transferred_ranges_v2	0
system	view_builds_in_progress	0
test_optimal	transactions	864000
test_optimal	user_profiles	864000
instacluster	recovery_codes	864000
instacluster	sla_latency	0
test_ttl	archived_logs	864000
test_ttl	cache_data	864000
test_ttl	session_data	864000
system_traces	events	0
system_traces	sessions	0
test_problematic	cdc_enabled	864000
test_problematic	default_settings	864000
test_problematic	high_bloom_fp	864000
test_problematic	low_index_intervals	864000



16 table(s) have gc_grace_seconds > 3 days or set to 0. This can lead to increased storage usage due to delayed tombstone cleanup and potential zombie reads.

1.15.1. Problematic Tables

Keyspace	Table	GC Grace Seconds
test_indexes	users_indexed	864000s
test_load	load_test_data	864000s
test_compaction	lcs_table	864000s
test_compaction	stcs_table	864000s
test_compaction	twcs_table	864000s
test_optimal	transactions	864000s
test_optimal	user_profiles	864000s
instacluster	recovery_codes	864000s
instacluster	sla_latency	0s
test_ttl	archived_logs	864000s
test_ttl	cache_data	864000s
test_ttl	session_data	864000s
test_problematic	cdc_enabled	864000s
test_problematic	default_settings	864000s
test_problematic	high_bloom_fp	864000s
test_problematic	low_index_intervals	864000s

1.15.2. Recommendations



- Review and reduce gc_grace_seconds for affected tables to 1-2 days (86400-172800 seconds) unless specific retention needs exist.
- Execute: ALTER TABLE keyspace.table WITH gc_grace_seconds = 86400;
- After changes, monitor tombstone counts with 'nodetool tablestats' and consider running 'nodetool cleanup' if needed.
- For tables with gc_grace_seconds=0, set an appropriate value to enable tombstone expiration.

1.16. Table Compression Settings Analysis

Checking compression configuration for all user tables in system_schema.tables.

keyspace_name	table_name	compression
test_indexes	users_indexed	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

system_auth	cidr_groups	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	cidr_permissions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	identity_to_role	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	network_permissions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	resource_role_permissions_index	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	role_members	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	role_permissions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_auth	roles	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	aggregates	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	column_masks	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

system_schema	columns	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	dropped_columns	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	functions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	indexes	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	keyspaces	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	tables	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	triggers	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	types	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_schema	views	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_load	load_test_data	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

test_compaction	lcs_table	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_compaction	stcs_table	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_compaction	twcs_table	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_distributed	parent_repair_history	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_distributed	partition_denylist	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_distributed	repair_history	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_distributed	view_build_status	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	IndexInfo	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	available_ranges	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	available_ranges_v2	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

system	batches	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	built_views	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	compaction_history	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	local	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	paxos	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	paxos_repair_history	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	peer_events	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	peer_events_v2	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	peers	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	peers_v2	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

system	prepared_statements	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	repairs	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	size_estimates	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	sstable_activity	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	sstable_activity_v2	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	table_estimates	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	top_partitions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	transferred_ranges	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	transferred_ranges_v2	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system	view_builds_in_progress	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

test_optimal	transactions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_optimal	user_profiles	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
instacluster	recovery_codes	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
instacluster	sla_latency	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_ttl	archived_logs	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_ttl	cache_data	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_ttl	session_data	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_traces	events	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
system_traces	sessions	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_problematic	cdc_enabled	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

test_problematic	default_settings	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_problematic	high_bloom_fp	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
test_problematic	low_index_intervals	{'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}



All 16 user table(s) have appropriate compression settings (LZ4Compressor enabled).

1.17. Row Cache Analysis

Checking row cache settings across all user tables. Row cache is often discouraged in production due to memory concerns.



All 16 user table(s) have row cache disabled.

keyspace_name	table_name	caching
test_indexes	users_indexed	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	cidr_groups	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	cidr_permissions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	identity_to_role	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	network_permissions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	resource_role_permissions_index	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	role_members	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	role_permissions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_auth	roles	{'keys': 'ALL', 'rows_per_partition': 'NONE'}

system_schema	aggregates	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	column_masks	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	columns	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	dropped_columns	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	functions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	indexes	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	keyspaces	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	tables	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	triggers	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	types	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_schema	views	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_load	load_test_data	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_compaction	lcs_table	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_compaction	stcs_table	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_compaction	twcs_table	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_distributed	parent_repair_history	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_distributed	partition_denylist	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_distributed	repair_history	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_distributed	view_build_status	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	IndexInfo	{'keys': 'ALL', 'rows_per_partition': 'NONE'}

system	available_ranges	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	available_ranges_v2	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	batches	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	built_views	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	compaction_history	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	local	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	paxos	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	paxos_repair_history	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	peer_events	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	peer_events_v2	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	peers	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	peers_v2	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	prepared_statements	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	repairs	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	size_estimates	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	sstable_activity	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	sstable_activity_v2	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	table_estimates	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	top_partitions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	transferred_ranges	{'keys': 'ALL', 'rows_per_partition': 'NONE'}

system	transferred_ranges_v2	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system	view_builds_in_progress	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_optimal	transactions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_optimal	user_profiles	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
instacluster	recovery_codes	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
instacluster	sla_latency	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_ttl	archived_logs	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_ttl	cache_data	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_ttl	session_data	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_traces	events	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
system_traces	sessions	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_problematic	cdc_enabled	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_problematic	default_settings	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_problematic	high_bloom_fp	{'keys': 'ALL', 'rows_per_partition': 'NONE'}
test_problematic	low_index_intervals	{'keys': 'ALL', 'rows_per_partition': 'NONE'}

1.18. Durable Writes Analysis

Checking durable_writes setting for all user keyspaces.



All user keyspaces have durable_writes enabled.

keyspace_name	durable_writes
test_indexes	True
system_auth	True

system_schema	True
test_load	True
test_compaction	True
system_distributed	True
system	True
test_optimal	True
instaclustr	True
test_ttl	True
system_traces	True
test_problematic	True

1.19. Materialized Views Analysis

Querying `system_schema.views` to list all materialized views, which can introduce performance overhead.



No user materialized views found.

1.20. User-Defined Functions and Aggregates Analysis

Querying `system_schema.functions` and `system_schema.aggregates` to list all UDFs and aggregates, flagging any using the 'java' language.



Query returned no results.



Query returned no results.



No user-defined functions or aggregates found.

1.21. Keyspace Replication Health Analysis

Analyzing replication strategies and factors for all user keyspaces.

keyspace_name	replication	durable_writes
test_indexes	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True

system_auth	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_schema	{'class': 'org.apache.cassandra.locator.L ocalStrategy'}	True
test_load	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
test_compaction	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_distributed	{'class': 'org.apache.cassandra.locator.Si mpleStrategy', 'replication_factor': '3'}	True
system	{'class': 'org.apache.cassandra.locator.L ocalStrategy'}	True
test_optimal	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
instacluster	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
test_ttl	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_traces	{'class': 'org.apache.cassandra.locator.Si mpleStrategy', 'replication_factor': '2'}	True
test_problematic	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True



All user keyspaces have healthy replication configurations (NetworkTopologyStrategy with RF >= 2).

1.22. Superuser Roles Analysis

Querying system_auth.roles to identify all superuser accounts for security review.

role	is_superuser
instacluster	True
icassandra	True

1.22.1. Superuser Roles Found

Role Name
instacluster
icassandra

1.22.2. Recommendations



- Review all superuser roles for least privilege principle: revoke superuser from accounts that don't need full admin access
- Use GRANT/REVOKE to assign specific permissions instead of superuser status
- Enable audit logging if available (cassandra.yaml: enabled: true) to track superuser actions
- Regularly audit role memberships: LIST ROLES; LIST PERMISSIONS ON ALL BY <role>
- Consider rotating passwords for superuser accounts and using strong, unique credentials

Chapter 2. Configuration

2.1. Keyspace Replication Strategy Analysis

Verifying that all user-defined keyspaces use NetworkTopologyStrategy and reporting replication factors per datacenter.

keyspace_name	replication	durable_writes
test_indexes	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_auth	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_schema	{'class': 'org.apache.cassandra.locator.L ocalStrategy'}	True
test_load	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
test_compaction	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
system_distributed	{'class': 'org.apache.cassandra.locator.Si mpleStrategy', 'replication_factor': '3'}	True
system	{'class': 'org.apache.cassandra.locator.L ocalStrategy'}	True
test_optimal	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True
instacluster	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.N etworkTopologyStrategy'}	True

test_ttl	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy'}	True
system_traces	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}	True
test_problematic	{'AWS_VPC_US_EAST_1': '3', 'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy'}	True



All 7 user keyspace(s) use NetworkTopologyStrategy.

2.1.1. Replication Factors per Datacenter

Keyspace	Datacenter	Replication Factor
test_indexes	AWS_VPC_US_EAST_1	3
test_load	AWS_VPC_US_EAST_1	3
test_compaction	AWS_VPC_US_EAST_1	3
test_optimal	AWS_VPC_US_EAST_1	3
instacluster	AWS_VPC_US_EAST_1	3
test_ttl	AWS_VPC_US_EAST_1	3
test_problematic	AWS_VPC_US_EAST_1	3

Chapter 3. Hardware



This check requires SSH access for Disk usage check.

Configure the following in your settings:

For single host: * `ssh_host`: Hostname or IP address

For multiple hosts (recommended for clusters): * `ssh_hosts`: List of hostnames/IPs

Authentication (required): * `ssh_user`: SSH username * `ssh_key_file` OR `ssh_password`: Authentication method

Optional: * `ssh_port`: SSH port (default: 22) * `ssh_timeout`: Connection timeout in seconds (default: 10)

3.1. AI-Generated Recommendations

Provides intelligent, context-aware recommendations based on dynamic analysis of database metrics.

3.1.1. AI Analysis Details

Parameter	Value
AI Provider	xAI
AI Model	grok-4
Prompt Size	64,767 characters (~16,191 tokens)
AI Processing Time	99.13 seconds

3.2. AI-Generated Recommendations

3.2.1. Executive Summary

The Cassandra 5.0.5 cluster exhibits a critical connectivity failure, with no peers detected in system.peers across multiple topology metrics, suggesting severe gossip or network misconfiguration in what appears to be a multi-node setup (3 nodes reported via Prometheus). This risks data unavailability, inconsistent reads/writes, and potential data loss. Resource metrics (heap <65%, CPU <10%, disk <1%, no pending compactions) are healthy, with low latencies (p95 read ~6ms, write ~4ms). Schema health shows high-priority issues in system_auth replication strategy (not using NetworkTopologyStrategy) and one table with gc_grace_seconds=0, increasing resurrection risks. Correlations: The connectivity failure likely causes uneven rack distribution (1:2 nodes across racks), amplifying failure risks in the single-DC setup with RF=3; no evidence of resource contention impacting latencies, but unresolved connectivity may mask underlying thread pool or compaction backlogs. Prioritize fixing connectivity before addressing schema optimizations.

3.2.2. Critical Issues

No Peers Detected in System.Peers - Severe Connectivity Failure

Operational Impact: Multiple topology metrics (datacenter summary, rack distribution, version distribution, overall topology) report no peers in system.peers, indicating nodes cannot communicate via gossip. This breaks replication, repairs, and streaming, leading to data inconsistency, unavailability, and potential loss in a multi-node cluster (Prometheus detects 3 nodes across 2 racks in 1 DC). Correlates with uneven rack distribution, where a rack failure could lose multiple replicas since distribution is imbalanced (us-east-1a:1, us-east-1b:2).

Action Steps: 1. Verify network configuration (listen_address, rpc_address, seeds) in cassandra.yaml on all nodes. 2. Check firewall rules and security groups for gossip ports (7000/7001). 3. Run `nodetool status` and `nodetool gossipinfo` on each node to diagnose. 4. If misconfiguration confirmed, update yaml files and perform a rolling restart.



Rolling restart required after config changes; schedule during maintenance window to avoid downtime. Monitor with `nodetool status` post-restart to confirm peers appear.

3.2.3. High Issues

System Auth Keyspace Not Using NetworkTopologyStrategy

Operational Impact: The system_auth keyspace uses a non-NTS strategy (reported as "None", likely SimpleStrategy), which is unsuitable even in single-DC setups as it lacks rack/DC awareness. This risks authentication unavailability if nodes in the same rack fail. Correlates with the critical connectivity issue, potentially exacerbating auth failures during partitions. All user keyspaces correctly use NTS with RF=3.

Action Steps: 1. Alter the keyspace to use NTS:

```
ALTER KEYSPACE system_auth WITH REPLICATION = {'class':  
'org.apache.cassandra.locator.NetworkTopologyStrategy', 'AWS_VPC_US_EAST_1': '3'};
```

1. Run `nodetool repair system_auth` cluster-wide to ensure consistency.
2. Verify with `DESCRIBE KEYSPACE system_auth;`



This change is cluster-wide; ensure all nodes are up before altering to avoid inconsistencies.

Uneven Rack Distribution in Single Datacenter

Operational Impact: With 3 nodes unevenly distributed (us-east-1a:1, us-east-1b:2) in a single DC using NTS RF=3, a failure in us-east-1b could lose 2 replicas, risking data loss despite RF=3. Correlates with no peers detected, indicating possible ongoing partition; single-DC setup lacks geographic redundancy, amplifying outage risks.

Action Steps: 1. Add nodes to balance racks (e.g., aim for 1 node per rack, or use 3 racks). 2. Use `nodetool move` or decommission/rebootstrap to redistribute tokens. 3. After balancing, run `nodetool cleanup` on affected keyspaces.



Node additions or moves require a maintenance window; perform rolling operations to maintain availability.

Table with `gc_grace_seconds=0` (No Tombstone Protection)

Operational Impact: Table 'sla_latency' in 'instaclustr' has `gc_grace_seconds=0`, allowing immediate tombstone garbage collection without grace period, risking deleted data resurrection on node failures or repairs. Other tables use 864000s (10 days), which is standard but should correlate with repair frequency; in a connectivity-impaired cluster, this heightens inconsistency risks.

Action Steps: 1. Set a safe value (e.g., 864000s):

```
ALTER TABLE instaclustr.sla_latency WITH gc_grace_seconds = 864000;
```

1. Schedule regular repairs (`nodetool repair`) to match grace period.
2. Monitor tombstones via `nodetool tablestats`.



Cluster-wide impact minimal, but repair after change to propagate.

3.2.4. Medium Issues

Non-Recommended Read Repair Settings on All Tables

Operational Impact: All 16 user tables have `read_repair_chance=0.0` and `dclocal_read_repair_chance=0.0`, disabling background consistency checks. This can lead to stale data in high-read workloads, especially with low latencies observed (no correlation to resource issues, but connectivity failure may hide inconsistencies).

Action Steps: 1. For each table, update to recommended:

```
ALTER TABLE <keyspace>.<table> WITH dclocal_read_repair_chance = 0.1;
```

1. Apply via script for all tables, then verify with `DESCRIBE TABLE`.
2. Monitor repair overhead post-change.

Secondary Indexes on 2 Columns

Operational Impact: 2 standard secondary indexes in 'test_indexes.users_indexed' (on 'email' and 'status') may cause performance overhead on writes and increase heap usage for index maintenance. Suitable only for low-cardinality columns; correlates with healthy heap/CPU, but could degrade if write load increases.

Action Steps: 1. Evaluate cardinality: Run queries to check unique values in indexed columns. 2. If high cardinality, replace with denormalized tables or materialized views:

```
CREATE MATERIALIZED VIEW <keyspace>.<view> AS SELECT * FROM <table> WHERE  
<indexed_col> IS NOT NULL PRIMARY KEY (...);
```

1. Drop indexes if unneeded: `DROP INDEX <index_name>;`.

High Bloom Filter False Positive Chance

Operational Impact: Table 'test_problematic.high_bloom_fp' has bloom_filter_fp_chance=0.15 (>0.1), increasing unnecessary disk reads and potentially read latency. No current latency issues, but correlates with secondary indexes if querying this table.

Action Steps: 1. Lower to 0.01:

```
ALTER TABLE test_problematic.high_bloom_fp WITH bloom_filter_fp_chance = 0.01;
```

1. Trigger compaction: `nodetool compact test_problematic high_bloom_fp`.
2. Monitor read metrics post-change.

Low Index Intervals on Table

Operational Impact: Table 'test_problematic.low_index_intervals' has min_index_interval=32, max=256 (lower than defaults 128/512), increasing memory usage for index summaries. Healthy heap usage currently, but could correlate with row cache if enabled (none here).

Action Steps: 1. Reset to defaults:

```
ALTER TABLE test_problematic.low_index_intervals WITH min_index_interval = 128 AND  
max_index_interval = 512;
```

1. Monitor JVM heap after change.

Multiple Superuser Roles

Operational Impact: 2 superuser roles ('instaclustr', 'iccassandra') exist, increasing security risks if compromised. In a connectivity-failed cluster, this could allow unauthorized access during recovery.

Action Steps: 1. Review and revoke unnecessary superuser privileges:

```
ALTER ROLE <role> WITH SUPERUSER = false;
```

1. Use least-privilege roles; disable defaults like 'cassandra' if present.

3.2.5. Low Issues

Single Datacenter Deployment

Operational Impact: Cluster operates in a single DC without geographic redundancy, risking total outage from DC failure. Correlates with uneven racks and connectivity issues, but resource metrics are healthy.

Action Steps: 1. Plan multi-DC expansion: Add a new DC, update keyspace replication to include it (e.g., add 'new_dc': '3' to NTS). 2. Use `nodetool rebuild` for data streaming to new DC.



Multi-DC addition requires maintenance; bootstrap new nodes sequentially.

No Low Issues in This Category

No additional low-severity issues identified beyond the above.