

1. Introduction

This review evaluates our current CloudWatch dashboards for Amazon Neptune against AWS' recommendations and the wider set of metrics exposed by Neptune in CloudWatch. The intent is to identify monitoring gaps, determine which metrics provide actionable insights, and make recommendations for improving our dashboards.

AWS emphasizes a small set of "most actionable" metrics, but Neptune actually publishes a broader set of metrics that we can leverage. Our focus is to build dashboards that highlight performance, reliability, and security while avoiding unnecessary noise.

2. Current Coverage

At present, our dashboards include:

- CPU Utilization
- Freeable Memory
- Database Connections
- Disk and Storage usage
- Network throughput

This gives us a basic view of system health but does not provide enough depth to understand query performance, replication health, or error patterns.

3. AWS Recommended Metrics – Gap Analysis

CPU Utilization

- Already monitored today.
- Value: Helps identify sustained CPU load and uneven distribution across replicas.
- Gap: We currently view cluster averages; we should expand to replica-level CPU metrics.

BufferCacheHitRatio

- Not monitored today.
- Value: If consistently below 99 percent, indicates that the workload exceeds memory capacity.
- Action: Should be added to dashboards.

MainRequestQueuePendingRequests

- Not monitored today.

- Value: If greater than zero, queries are being queued, meaning the instance is under pressure.
- Action: Should be added to dashboards.

Freeable Memory

- Already monitored today.
- Value: AWS advises this is not highly actionable.
- Action: Keep visible but remove from alerting thresholds.

OpenCypherBoltOpenConnections and GremlinWebSocketOpenConnections

- Not monitored today.
- Value: Useful only if our applications use Bolt or Gremlin. Can detect potential connection leaks.
- Action: Add if relevant.

GremlinClientErrorsPerSec, NumOpenCypherClientErrorsPerSec, and TotalClientErrorsPerSec

- Not monitored today.
- Value: Baseline client errors and alert on abnormal spikes.
- Action: Add to dashboards.

Event Subscriptions

- Not enabled today.
- Value: Provides proactive notification for failovers, backups, and maintenance.
- Action: Should be enabled and integrated into CloudWatch.

4. Additional Native Neptune Metrics Available in CloudWatch

Beyond AWS' short list, Neptune provides many other metrics that we can use today without custom instrumentation. These offer deeper operational visibility.

Replication and Freshness

- **ClusterReplicaLag**, **ClusterReplicaLagMaximum**, **ClusterReplicaLagMinimum** give insight into how current replicas are relative to the writer.

Concurrency and Transactions

- **NumTxOpened**, **NumTxCommitted**, and **NumTxRolledBack** provide visibility into transaction flow and failure patterns.
- **NumQueuedRequestsPerSec** shows how many requests are stuck waiting.

Protocol and Error Insights

- **GremlinRequestsPerSec**, **OpenCypherRequestsPerSec**, and **SparqlRequestsPerSec** highlight query throughput by API.
- Corresponding error metrics such as **GremlinServerErrorsPerSec** and **OpenCypherServerErrorsPerSec** allow us to pinpoint failing paths.
- **TotalRequestsPerSec** and **TotalClientErrorsPerSec** provide an aggregate cluster-level view.

Cache Effectiveness

- **BufferCacheHitRatio** is the primary signal for memory effectiveness.
- For Gremlin, additional cache metrics such as **NumResultCacheHit**, **NumResultCacheMiss**, and **ResultCacheSizeInBytes** can be used to measure cache utilization.

Storage and I/O

- **VolumeReadIOPs**, **VolumeWriteIOPs**, and **VolumeBytesUsed** highlight storage activity.
- **TempStorageIOPS** and **TempStorageThroughput** show whether the system is spilling to temp storage.
- **StorageNetworkThroughput** and **NetworkThroughput** provide network activity data.

Backup and Cost Visibility

- **BackupRetentionPeriodStorageUsed**, **SnapshotStorageUsed**, and **TotalBackupStorageBilled** show backup footprint and cost trends.

Global Database

- **GlobalDbProgressLag**, **GlobalDbDataTransferBytes**, and **GlobalDbReplicatedWriteIO** allow us to measure replication performance across regions.

5. Metrics to Deprioritize

Not every metric is actionable. To keep dashboards clean:

- Freeable Memory should remain visible but not used for alerting.
 - Generic network in and out metrics should not be primary alert sources since protocol-level metrics are more actionable.
 - Unstructured error counts should be avoided; structured error metrics by protocol are more useful.
-

6. Recommendations

1. Add BufferCacheHitRatio and MainRequestQueuePendingRequests immediately, as they are highly actionable.
 2. Enable Event Subscriptions for failover, backup, and maintenance events.
 3. Add client and server error metrics for Gremlin and openCypher.
 4. Expand dashboards with replication, transaction, and I/O metrics to move from “basic health” to “deep operational visibility.”
 5. Retain Freeable Memory and network metrics only for secondary visibility, not as alert triggers.
 6. Document thresholds and corrective actions in Confluence to ensure every alert is tied to a runbook.
-