NeoStats Data Engineering Use Case – Detailed Documentation

Virtual Server Monitoring & Performance Optimisation

Prepared by: **P Vikram**

Date: 26-11-2025

---

1. Introduction

This document explains the complete end-to-end solution developed for the NeoStats Data Engineering Use Case, focusing on virtual server monitoring and performance optimisation. The objective was to ingest and clean server performance data, combine logs from two stations, engineer key metrics such as Availability %, enrich with server metadata, build a curated dataset, and create a Power BI dashboard for insights.

2. Problem Summary

NeoStats Analytics monitors multiple virtual servers across various clusters and locations. Server performance logs come from two different stations containing missing values and inconsistent fields. The goal was to build a clean unified dataset and a monitoring dashboard showing CPU utilisation, availability, and downtime.

3. Dataset Description

- Server_Metadata (100 rows)
- Contains Server_ID, Hostname, OS, Location, Cluster, Admin details.
- Server_Performance_Station1 (3000 rows)
- Performance metrics plus extra irrelevant fields.
- Server_Performance_Station2 (2000 rows)
- Performance metrics only.

4. Solution Architecture

Excel File → Python (Cleaning & Transformation) → Join Metadata → Curated CSV → Power BI Dashboard

5. Technology Stack

Python (Pandas, NumPy), Jupyter Notebook, CSV File Storage, Power BI Desktop.

6. Data Pipeline Steps

Step 1 — Ingestion

Loaded all sheets using pandas.read_excel().

Step 2 — Cleaning

Removed irrelevant columns: Config_Version, Last_Patch_Date, and Deployment_Token.

Added Source_Station column.

Aligned columns.

Step 3 — Combine Logs

Union of Station1 and Station2.

Converted Log_Timestamp into Log_Date & Hour_Of_Day.

Step 4 — Null Imputation

Imputed numeric fields with:

• Median per Server_ID

• Global median fallback

Step 5 — Feature Engineering Created:

• Total_Time_Hours = Uptime + Downtime

• Availability (%) = (Uptime / Total_Time_Hours) × 100

• High CPU/Memory/Disk Flags (>85%)

Step 6 — Metadata Join

Left join on Server_ID.

Filled missing metadata with “Unknown”.

Step 7 — Export Curated Dataset

Saved as curated_server_performance.csv.

7. Data Model Design

Fact Table: Performance logs + KPIs.

Dimension Table: Server_Metadata. **Relationship:** Server_ID.

8. Power BI Dashboard

Page 1 — Overview

• KPI Cards (Avg CPU, Avg Availability, Total Downtime)

• Slicers (Cluster, Location)

• Bar Chart (Downtime by Location)

• Line Chart (Avg CPU over Time)

Page 2 — Server Drilldown

Table with Server_ID, Location, Cluster, Avg CPU, Avg Memory, Avg Availability, and Total Downtime.

9. Key Insights

• Singapore has the highest downtime (~1749 hours).

• Average CPU utilization is ~50-55%.

• CPU trend shows dips and rises, indicating workload changes.

• Most servers have >98% availability.

• Cluster CL-7 shows slightly higher CPU usage.

10. Conclusion

This project successfully demonstrates an end-to-end data engineering workflow: ingestion, cleaning, transformation, metadata enrichment, dataset creation, and dashboard reporting. The final output provides clear insights into server performance and helps in operational monitoring.

11. Deliverables

• server_pipeline.ipynb

• NeoStats_Server_Dashboard.pbix

• NeoStats_Detailed_Documentation.pdf

• curated_server_performance.csv