Business Requirements Document

Project name: Linux Cluster Monitoring Agent

Project manager: Trevor Doto

Date submitted: May 26, 2025

Document author: Aditya Khajanchi

Document status: ☐ Draft ☒ Proposed☐Validated ☐Approved

Document version: 1.0

Table of Contents

1. Executive summary
2. Project objectives
3. Needs Statement
4. Project Scope
    4.1. In-scope
    4.2. Out-of-scope
5. Requirements
    5.1. Functional Requirements
    5.2. Non-functional Requirements
6. Key Stakeholders
7. Project Timeline
8. Cost-Benefit Analysis

REVISIONS

Name	Date	Reason	Version
Aditya Khajanchi	June 2, 2025	Proposed SRD	v1.0

1. **Executive summary**
   1.1. This BRD outlines the requirements for an automated monitoring solution to be delivered by June 17, 2025, that will record hardware specifications on initial installation and collect real-time CPU, memory, and disk I/O metrics from up to 100 Linux servers. All data will be stored in a Docker-containerized PostgreSQL database, with scripts version-controlled on GitHub and setup instructions in a README file. An MVP will be tested on a single node before scaling cluster-wide. Replacing ad-hoc data pulls, this solution reduces manual effort, enables trend analysis and capacity planning, and accelerates troubleshooting.

2. **Project objectives**
   2.1. Deliver a monitoring solution by June 17, 2025, for a cluster of up to 100 Linux servers that captures hardware specifications and performance data, to support future resource planning and report generation.

3. **Needs Statement**
   3.1. The existing process for data collection and resource management relies on occasional, ad-hoc data pulls and manual report generation. This ad-hoc approach could cause inconsistent data quality, delayed detection of performance bottlenecks, and leaves the team unable to analyze trends over time or accurately forecast capacity needs across the cluster. Hence, the LCA Team at Jarvis requires a product to automatically capture hardware specifications and real-time usage metrics from running Linux servers to monitor resource utilization and make informed decisions about scaling infrastructure as

4. **Project Scope**
   4.1. In scope:
      4.1.1. Recording hardware specifications of each node on initial installation
      4.1.2. Real-time collection of usage metrics
      4.1.3. Storage of hardware specifications and usage metrics in PostgreSQL database
      4.1.4. Docker based deployment of the PostgreSQL database
      4.1.5. Testing of MVP on one Linux server
      4.1.6. Software version control via Git and hosted through GitHub
      4.1.7. README.md file with project documentation and quick start guide
      4.1.8. Conducting a staff training and Q&A session as part of project delivery
   4.2. Out of scope:
      4.2.1. Re-collection of hardware data after upgrades

4.2.2. Development of performance dashboards or visualizations
4.2.3. Implementation of real-time usage alerts or notifications
4.2.4. Integration with existing infrastructure or external systems
4.2.5. Configuration of firewall rules or network connections

5. Requirements

5.1. Functional Requirements

5.1.1. The system shall collect hardware specifications including CPU details, memory, disk I/O from each node.
5.1.2. The system shall collect real-time resource usage metrics including CPU usage, memory consumption, disk I/O from each node.
5.1.3. The system shall run the monitoring script automatically at every minute using crontab.
5.1.4. The system shall store the collected data in a PostgreSQL database, provisioned using Docker.
5.1.5. All scripts and configurations shall be version-controlled using Git and hosted on GitHub.

5.2. Non-functional Requirements

5.2.1. The solution shall support the addition or removal of nodes.
5.2.2. The solution shall be compatible with Rocky Linux 9 servers.
5.2.3. Persisted data shall be able to support SQL queries to retrieve historical and current performance data for analysis.
5.2.4. All scripts shall be tested and validated manually against test data.
5.2.5. MVP to be tested on a single node but shall support a cluster of Linux servers.

6. Key Stakeholders

Role	Responsibility
Project Manager	• Ensure timely execution by holding all parties accountable to the project timeline. • Coordinate communication between stakeholders.

LCA Team	• Provide business requirements for the project. • Participate in feedback sessions to validate developments. • Pay for costs associated with the project development including staff training.
Development team	• Develop a robust, tested, and error-free solution for the client. • Address technical challenges and incorporate stakeholder feedback.

7. Project Timeline

Phase	**Duration**	**Milestone**
Requirements Gathering & Team Assignments	2 Days	Finalize project requirements, scope, and work distribution
Development	7 Days	Setup Git repository, write bash scripts, automate with cron, version control
Testing	2 Days	Validate functionality using test data
Documentation	2 Days	Create README.md file with project information and quick start guide
Revisions and Delivery	3 Days	Apply feedback, finalize code, and deliver project
Total Duration	16 Days	

8. Cost-Benefit Analysis

8.1. Benefits from the new system:

8.1.1. Enables real-time resource monitoring through automated background data collection.

8.1.2. Supports historical data analysis for forecasting resource usage and planning capacity.

8.1.3. Faster server performance troubleshooting and root cause analysis.

8.1.4. Scales efficiently to monitor up to 100 Rocky Linux servers.

8.1.5. Facilitates future functionalities upgrade ready with the help of documentation and version control.

8.2. Reduction in:

8.2.1. System downtime due to unmonitored resource bottlenecks.

8.2.2. Operational efforts and potential human errors in data collection and monitoring.

End of Document