

# Integration of Airflow and Grafana for Enhanced Monitoring of DAGs: A Case Study

Dr Vinit Kumar Lohan<sup>1</sup> and Sourav Parmar<sup>2</sup>

<sup>1</sup>Assistant Professor, CSE Department, VCE Rohtak

<sup>2</sup>Student and Member, Research Group, Rohtak

**Publishing Date: June 25, 2018**

## Abstract

This paper presents the integration of Apache Airflow with Grafana to enhance monitoring capabilities for Directed Acyclic Graphs (DAGs), addressing the limitations of Airflow's default monitoring tools. By migrating the database from SQLite to MySQL and developing a custom Grafana dashboard, the project achieves real-time, detailed visualizations of DAG statuses, including successes, failures, and performance metrics. This integration significantly improves operational oversight, efficiency, and the user experience in workflow management, demonstrating a practical approach to leveraging advanced visualization tools for better monitoring and analysis of complex workflows.

**Keywords:** *Airflow, Grafana, DAG, Workflow, Integration Process.*

## 1. Introduction

In the evolving landscape of data engineering and workflow management, the ability to efficiently schedule, monitor, and manage complex workflows is paramount. Apache Airflow has emerged as a leading open-source platform, offering robust capabilities for orchestrating complex computational workflows, data processing jobs, and ETL tasks. By defining workflows as Directed Acyclic Graphs (DAGs), Airflow allows for flexible, scalable, and manageable workflow automation. However, despite its comprehensive scheduling and monitoring capabilities, Airflow's native monitoring tools often fall short in delivering the granularity and visual clarity required for optimal operational oversight.

The significance of real-time monitoring in workflow management cannot be overstated. It enables data engineers and DevOps teams to swiftly detect failures, assess workflow

performance, and ensure the reliability and efficiency of data processing pipelines. In this context, the integration of Apache Airflow with Grafana, a leading open-source platform for monitoring and observability, presents an innovative solution. Grafana offers advanced data visualization capabilities that can significantly enhance the monitoring of Airflow DAGs, providing a more intuitive and informative overview of workflow statuses, successes, and failures.

This thesis explores the integration of Airflow with Grafana to address the limitations of Airflow's native monitoring tools. The primary objective is to develop a comprehensive monitoring solution that leverages Grafana's advanced visualization capabilities to provide real-time insights into the status of Airflow DAGs. By migrating Airflow's backend database from SQLite to MySQL, this project aims to lay the groundwork for robust data handling and connectivity between Airflow and Grafana. The ultimate goal is to enhance the user experience by offering a detailed and interactive dashboard that not only displays the current status of workflows but also enables users to drill down into specific DAGs to view logs and detailed execution metrics.

Through this integration, the thesis aims to achieve the following objectives:

- To overcome the limitations of Airflow's default monitoring tools by providing a more detailed and visually intuitive overview of DAG statuses.
- To enable real-time monitoring and alerts for workflow successes, failures, and performance metrics, thereby improving operational efficiency and reliability.

- To enhance the usability of monitoring tools by creating a seamless link between Grafana dashboards and Airflow logs, facilitating quick access to detailed execution information.

In addressing these objectives, this thesis not only contributes to the field of data engineering and workflow management by improving the operational oversight of Airflow workflows but also exemplifies the practical application of integrating disparate technology stacks to solve specific industry challenges.

## 2. Methodology

The methodology section of a thesis outlines the specific processes, tools, and procedures used to achieve the project's objectives. In the context of integrating Apache Airflow with Grafana for enhanced DAG monitoring, this section delves into the system design, database migration, integration process, and the development of a custom dashboard, as well as addressing challenges encountered and their solutions.

### 2.1 System Design

#### 2.1.1 Initial Conditions and Requirements

The project commenced with an assessment of the existing monitoring capabilities within Apache Airflow, which relied on its internal database powered by SQLite. The primary requirement was to enhance monitoring visibility and access to detailed logs for running, successful, and failed DAGs in a more user-friendly and informative manner than what the default Airflow interface offered. The solution needed to be scalable, accessible, and capable of providing real-time insights into workflow operations.

#### 2.1.2 Database Migration: SQLite to MySQL

A critical initial step was migrating Airflow's database from SQLite to MySQL. This migration was necessitated by SQLite's limitations in handling concurrent accesses and its lack of support for real-time data processing and visualization, which are essential for robust workflow monitoring. MySQL was chosen for its scalability, reliability, and broader support for complex queries and operations. The migration

process involved exporting existing data from SQLite, transforming it to be compatible with MySQL, and importing it into the new MySQL database.

### 2.2 Integration Process

#### 2.2.1 Connecting Airflow with Grafana

The integration of Airflow with Grafana required establishing a direct connection between the MySQL database and Grafana. This was achieved by configuring Grafana to recognize the MySQL database as a data source, enabling Grafana to fetch and visualize data directly from Airflow's operational database. This step was crucial for real-time monitoring and visualization.

#### 2.2.2 Custom Dashboard Development

With the data connection established, the next step was developing custom dashboards in Grafana tailored to the monitoring needs of Airflow DAGs. These dashboards were designed to display key metrics such as DAG run statuses, execution times, and success/failure rates, with additional features like alerts for failures and clickable links to view detailed logs. The development process involved selecting appropriate visualization widgets and configuring them to query and display data dynamically from the MySQL database.

#### 2.2.3 Challenges and Solutions

Throughout the project, several challenges were encountered and subsequently addressed:

- **Data Compatibility and Loss:** The migration from SQLite to MySQL posed risks of data compatibility issues and potential data loss. A meticulous data export/import procedure, coupled with thorough testing, was implemented to mitigate these risks.
- **Real-Time Data Visualization:** Ensuring that Grafana could display data from Airflow in real-time required optimizing MySQL queries and Grafana refresh settings. This was achieved through trial and error, optimizing query performance and dashboard settings.
- **Usability and Accessibility:** Designing dashboards that were both informative and user-friendly was paramount. Feedback from potential users was solicited early in the design process to guide the layout and functionality of the dashboards.
- **Learning Curve:** The project team had to rapidly improve their Grafana dashboard

design and MySQL administration. This was overcome by leveraging online resources, community forums, and extensive documentation available for both platforms.

### 3. Implementation

The implementation phase of this project was pivotal in bringing the theoretical design and planning stages into a functional, real-world application. This phase encompassed the technical details of database migration, the configuration of the Grafana dashboard, the visualization of Directed Acyclic Graphs (DAGs) statuses, alert setups for success and failure notifications, linking to logs for in-depth analysis, and refining user interaction and experience.

#### 3.1 Database Migration Details

The migration from SQLite to MySQL was a foundational step in enhancing the monitoring capabilities of Apache Airflow with Grafana. The process involved several key steps:

- **Data Export from SQLite:** Utilizing tools to extract the data from the SQLite database into a format that could be easily imported into MySQL.
- **Schema Conversion:** Adjusting the database schema to be compatible with MySQL, including data types and index configurations.
- **Data Import to MySQL:** Importing the converted data into MySQL, ensuring integrity and consistency.
- **Airflow Configuration:** Updating Airflow's configuration settings to utilize MySQL as its backend database, including adjusting connection strings and testing connectivity.

#### 3.2 Grafana Dashboard Configuration

With the data source successfully migrated and connected, configuring the Grafana dashboard was the next critical task. This involved:

- **Data Source Setup:** Adding the MySQL database as a data source in Grafana.
- **Dashboard Creation:** Designing a dashboard layout that would effectively display the key metrics of Airflow DAGs.
- **Query Configuration:** Writing SQL queries for each panel to fetch and display data, such as DAG run times, success/failure rates, and execution logs.

#### 3.3 Visualizing DAG Statuses

The core feature of the Grafana dashboard was the visualization of DAG statuses. Panels were

designed to provide real-time insights into the state of each DAG, including:

- **Running:** Current active DAGs.
- **Success:** Recently succeeded DAGs.
- **Failed:** DAGs that have failed in the last run.

These visualizations utilized color coding and graphs to provide at-a-glance insights, enhancing the monitoring experience.

#### 3.4 Alert Setup for Success/Failure

To promptly notify users of critical events, alert rules were configured within Grafana based on specific conditions:

- **Failure Alerts:** Notifications were set up to trigger when any DAG failed, sending alerts through configured channels like email or Slack.
- **Success Notifications:** While less critical, alerts for successful completions were also established for key workflows, ensuring stakeholders were informed.

#### 3.5 Linking to Logs for Detailed Analysis

A significant enhancement to the dashboard was the ability to link directly to detailed logs for each DAG. This feature allowed users to click on a DAG status within the dashboard and be taken directly to the corresponding logs in Airflow for in-depth analysis, significantly speeding up troubleshooting and review processes.

#### 3.6 User Interaction and Experience

Throughout the implementation phase, user interaction and experience were focal points. Feedback was solicited from actual users to refine the dashboard design, ensuring that it was not only functional but also intuitive. User guides and documentation were created to assist users in navigating the dashboard and understanding the metrics presented.

This project's implementation successfully showed how integrating Airflow with Grafana could significantly enhance workflow monitoring, providing a more informative, interactive, and user-friendly monitoring solution. Through careful planning and execution, the project addressed the initial limitations of Airflow's monitoring capabilities, offering a robust tool for data engineers and DevOps teams.

## 4. Results and Evaluation

The integration of Apache Airflow with Grafana for enhanced DAG monitoring has undergone a thorough evaluation phase to assess its impact on

system performance, user experience, and overall efficiency compared to Airflow's default monitoring capabilities. This section outlines the project's outcomes in terms of system performance analysis, user feedback and usability, and a comparative overview with the default Airflow monitoring tools.

#### 4.1 System Performance Analysis

The migration of the database from SQLite to MySQL and the subsequent integration with Grafana were critical steps that aimed to not only enhance monitoring capabilities but also ensure system performance was not adversely affected. The following key points were noted in the system performance analysis:

- **Query Response Time:** The response times for queries made by Grafana to the MySQL database showed significant improvement over SQLite, indicating efficient data retrieval for monitoring purposes.
- **Dashboard Load Times:** Initial concerns regarding the potential increase in dashboard load times were mitigated through optimized SQL queries and Grafana settings, resulting in minimal impact on performance.
- **Resource Utilization:** Monitoring the system's resource utilization under the new setup showed a moderate increase in database resource usage, which was within acceptable limits and justified by the enhanced capabilities.

#### 4.2 User Feedback and Usability

Feedback from users who interacted with the Grafana dashboard was overwhelmingly positive, highlighting several aspects:

- **Intuitiveness:** Users found the Grafana dashboard to be more intuitive and visually appealing compared to Airflow's native interface, making it easier to understand the status of DAGs at a glance.
- **Functionality:** The ability to drill down into specific DAGs to access detailed logs directly from the dashboard was highly praised, as it significantly streamlined the troubleshooting process.
- **Alerting Features:** The customizable alerting feature was noted for its flexibility and effectiveness in notifying users about DAG failures or successes, enhancing operational awareness and response times.

#### 4.3 Comparison with Default Airflow Monitoring

The project's success was further underscored by a

direct comparison with Airflow's default monitoring tools, revealing several areas of improvement:

**Visualization and Accessibility:** The Grafana dashboard offered superior visualization capabilities, with dynamic graphs and color-coded statuses providing a more accessible overview of DAG states.

- **Real-Time Monitoring:** Real-time monitoring and alerting capabilities were markedly improved, offering immediate insights into DAG performances and issues as they happened.

- **Customization and Scalability:** Grafana provided a higher degree of customization and scalability, allowing for the monitoring setup to be tailored to specific needs and easily expanded as system complexity grows.

In conclusion, the integration of Apache Airflow with Grafana for DAG monitoring represents a significant advancement in the management and oversight of workflow operations. The positive outcomes observed in system performance, enhanced user experience, and the comparative benefits over default Airflow monitoring tools validate the project's objectives and approach. This integration not only addresses the immediate needs for better monitoring solutions but also sets a foundation for further explorations and improvements in workflow management technologies.

### 5. Discussion

The integration of Apache Airflow with Grafana has yielded significant insights and findings, underscoring the potential for advanced monitoring tools to enhance workflow management. This section delves into the project's key insights, its implications for the broader field of workflow management, and acknowledges the limitations encountered during the study.

#### 5.1 Insights and Findings

One of the primary insights from this project is the critical role of visualization and real-time monitoring in managing complex workflows. The use of Grafana dashboards demonstrated a notable improvement in the ability to quickly assess and respond to workflow states, underscoring the value of intuitive visual tools in operational environments. Furthermore, the project highlighted the importance of database choice in supporting scalable and efficient data handling for monitoring purposes. The migration from SQLite to MySQL facilitated a

more robust integration with Grafana and improved overall system performance.

Another key finding was the effectiveness of customized alerting mechanisms in enhancing operational responsiveness. By tailoring alerts for

specific DAG outcomes, teams can prioritize issues and reduce downtime, significantly impacting workflow reliability and efficiency.

## 5.2 Implications for Workflow Management

The results of this project have several implications for the field of workflow management. Firstly, they reinforce the necessity of adopting flexible and powerful monitoring tools that can adapt to the complexities of modern data workflows. This integration serves as a case study for how such tools can be leveraged to improve oversight and operational efficiency.

Secondly, this project illustrates the potential for open-source tools to provide enterprise-level monitoring solutions, democratizing access to advanced technology for organizations of all sizes. The ability to customize and extend these tools can empower teams to build monitoring solutions that precisely fit their operational needs, fostering innovation and continuous improvement.

Lastly, the project underscores the ongoing need for technical skills development in areas such as database management, data visualization, and system integration. As the complexity of workflow systems grows, so too does the demand for skilled professionals capable of navigating and optimizing these environments.

## 5.3 Limitations of the Study

While the project achieved its primary objectives, it was not without limitations. One of the main constraints was the scope of testing environments. The implementation and evaluation were conducted within a controlled setting, which may not fully replicate the complexities and scale of production environments in diverse organizations.

Another limitation lies in the focus on specific technologies (Airflow, MySQL, and Grafana). While these tools are widely used, the findings may not be directly applicable to organizations utilizing different workflow management or monitoring solutions.

Furthermore, the project did not extensively explore the potential security implications of the integration, particularly in terms of data access and management. Future work could benefit from a deeper examination of these aspects to ensure the secure deployment of similar integrations.

## 6. Conclusion

The integration of Apache Airflow with Grafana undertaken in this thesis has successfully demonstrated a significant enhancement in the monitoring and management of Directed Acyclic Graphs (DAGs), addressing key limitations of Airflow's default monitoring capabilities. This project not only contributed to the field of workflow management by providing a detailed methodology for improving operational oversight but also showcased the practical application and benefits of utilizing advanced data visualization tools in complex workflow environments.

### 6.1 Summary of Contributions

The core contributions of this thesis include:

- **Enhanced Monitoring Capabilities:** By integrating Grafana with Airflow, this project has introduced a superior monitoring solution that offers real-time insights, intuitive visualization, and comprehensive analysis of DAG statuses, execution times, and outcomes.
- **Database Migration and Integration:** The migration from SQLite to MySQL as part of this integration effort has underscored the importance of database choice in supporting scalable and efficient data handling for advanced monitoring purposes.
- **Customizable Alerting Mechanism:** The development of a customizable alerting system within the Grafana dashboard has proven effective in enhancing operational responsiveness and reducing downtime, thereby improving the overall reliability of workflow processes.
- **Improved User Experience:** The intuitive and user-friendly interface of the Grafana dashboard has significantly improved the user experience, making it easier for teams to monitor, analyze, and troubleshoot workflows.

### 6.2 Future Work Directions

While this project has made notable strides in enhancing DAG monitoring within Apache Airflow, several areas for future work have been identified:

- **Security Enhancements:** Further research and development could focus on enhancing the security aspects of the integration, especially concerning data access, management, and privacy within the

Grafana dashboard.

- **Scalability and Performance Optimization:**

As organizations continue to scale their operations, future efforts could explore more advanced database solutions and optimization techniques to ensure the monitoring system remains efficient and scalable.

- **Broader Tool Integration:** Investigating the integration of Airflow with other monitoring and data visualization tools could provide additional flexibility and functionality to meet the diverse needs of different organizations.

- **AI and Machine Learning for Predictive Analytics:** Incorporating artificial intelligence and machine learning to analyze workflow patterns and predict potential failures or bottlenecks before they occur could represent a significant advancement in proactive workflow management.

- In conclusion, this thesis lays a foundation for future innovations in workflow monitoring and management. By addressing immediate challenges and exploring new technologies and methodologies, there is a vast potential for further improving the efficiency, reliability, and scalability of workflow systems.

Grafana Labs. (2017). *Grafana documentation*.

Retrieved from <https://grafana.com/docs/>

## References

- [1] Restack. (2018). [Online], Airflow knowledge: Apache + Grafana. Retrieved from <https://www.restack.io/docs/airflow-knowledge-apache-grafana>
- [2] R. Perkasa. (2018). [Online] , "Monitoring Airflow Metrics with Grafana," from: <https://medium.com/@perkasaid.rio/monitoring-airflow-metrics-with-grafana-29ebb43100a3>
- [3] HostedMetrics, (2018) [Online]. from <https://hostedmetrics.com/use-with/airflow/>
- [4] Thalheim, J., Rodrigues, A., Akkus, I. E., Bhatotia, P., Chen, R., Viswanath, B., Jiao, L., & Fetzer, C. (2017). Sieve: Actionable Insights from Monitored Metrics in Microservices. arXiv:1709.06686.
- [5] Röhl, T., Eitzinger, J., Hager, G., & Wellein, G. (2017). *LIKWID monitoring stack: A flexible framework enabling job specific performance monitoring for the masses*. arXiv preprint arXiv:1708.01476.
- [6] Grafana Labs. (2017). *Grafana documentation*. Retrieved from <https://grafana.com/docs/>