International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Ansible Upgrade in Mission-Critical Systems: Ensuring Backward Compatibility and Role Integrity

Pathik Bavadiya

Submitted:15/11/2023 **Revised:** 27/12/2023 **Accepted:** 05/01/2024

Abstract: In mission-critical systems, where it is necessary to ensure uninterrupted operations during upgrades, Ansible is commonly used for automating configuration management through the use of configuration management automation. The purpose of this study was to evaluate the influence that updating versions of Ansible has on backward compatibility and role integrity within environments like an environment. A descriptive and exploratory research design was utilized in order to assess the data collected from five different real-world information technology infrastructures as well as controlled upgrade testing. According to the findings of the study, small version upgrades typically maintained compatibility and role functionality, however bigger version leaps frequently resulted in role failures and necessitated rollbacks. Among the primary reasons for upgrade problems were the presence of obsolete modules and changes in role syntax. Qualitative insights from DevOps professionals emphasized the necessity of comprehensive testing, detailed documentation, automated rollback mechanisms, and version pinning to ensure upgrade success. For the purpose of properly managing Ansible upgrades and reducing disruptions to important automation activities, the findings provide valuable suggestions.

Keywords: *Ansible upgrade, backward compatibility, role integrity, mission-critical systems, automation, rollback, DevOps best practices.*

1. Introduction

It is now possible for businesses to deploy, configure, and maintain complex systems at scale with minimal human interaction thanks to automation, which has formed the backbone of modern IT infrastructure management. Because of its agentless architecture, user-friendliness, and strong role-based configuration management, Ansible stands out among the many automation tools that are now available. As a result, it has gained widespread adoption in a variety of contexts that are considered to be mission-critical. Backward compatibility and the maintenance of role integrity are two issues that are brought up by frequent updates, which are a consequence of Ansible's ongoing development. Mission-critical systems require a high level of availability and dependability, because there is very little space for errors or disruptions throughout the upgrading process. Significant operational setbacks can be caused by any incompatibility or failure in automation roles, which can have an impact on both the supply of services and the continuation of corporate operations. Therefore, it is necessary for companies that rely heavily on these technologies to have a solid understanding of how Ansible upgrades affect the automated workflows that are already in place and to establish solutions that can mitigate the risks that are connected with these upgrades. The purpose of this study is to investigate these problems by investigating the technical effects on backward compatibility, reviewing real-world upgrade scenarios, and evaluating the integrity of roles once an update has been performed. In addition to this, it incorporates the experiences of DevOps specialists in order to highlight best practices that guarantee a smooth transition between different versions of Ansible without sacrificing the automation services that are critically important to the mission. The purpose of this research is to give a thorough framework for properly managing Ansible updates in high-stakes information technology environments through the methodology of this study.

1.1. Background of the study

In the quickly changing environment of information technology that we are currently experiencing, automation technologies such as Ansible play a crucial role in the effective and dependable

Vice President, Production Services (Independent Researcher) BNY, New York, USA

pathikbavadiya1900@gmail.com

ORCID: 0009-0003-4405-3657

management of mission-critical and complex infrastructure. Because of its ease of use, agentless design, and modular roles, Ansible has become the go-to solution for configuration management and orchestration across a wide variety of contexts, such as hybrid infrastructures, on-premise data centers, and cloud environments. Frequent updates and version upgrades of Ansible, on the other hand, provide substantial issues. This is especially true in mission-critical systems, where even modest disruptions can result in costly downtime or service outages. In order to keep existing automation workflows running without requiring substantial rewrites or failures, it is essential to ensure that backward compatibility is maintained during these upgrades. It is similarly important to maintain role integrity, which refers to the correct operation of stated automation roles, because any breach in this integrity might lead to larger system instability. There is a lack of thorough research that provides a systematic evaluation of the technical consequences of Ansible updates on backward compatibility and role performance in real-world mission-critical environments. This is despite the fact that the relevance of this topic is well recognized. This gap necessitates a focused investigation to identify common pitfalls, assess upgrade outcomes, and develop practical strategies for managing Ansible upgrades effectively. It is possible for enterprises to better protect their operational continuity while simultaneously harnessing the benefits of the most recent Ansible features and upgrades if they undertake these issues and address them.

1.2. Upgrade Challenges in Mission-Critical Ansible Systems

Upgrading Ansible in mission-critical systems presents a number of important problems that have the potential to affect the stability of the system as well as the reliability of the automation. Because these settings require continuous operation, any faults or incompatibilities that are due to upgrades could possibly result in significant financial losses. Managing dependencies across a wide variety of infrastructure components, guaranteeing backward compatibility with old playbooks and roles, and dealing with obsolete modules and syntax changes are all examples of common issues. Furthermore, the limited testing options and the complexity of automation workflows both contribute to an increase in the likelihood of errors that are not discovered. It is necessary to engage in careful planning, comprehensive validation, and powerful rollback methods in order to address these problems and ensure that operations continue to run smoothly both during and after upgrades.

Key Upgrade Challenges:

- Maintaining backward compatibility with legacy playbooks and roles
- Managing deprecated modules and changes in Ansible syntax
- Handling dependency and environment version conflicts
- Testing upgrades in complex, diverse infrastructure environments
- Mitigating risks of automation failures impacting mission-critical operations
- Implementing reliable rollback and recovery strategies

1.3. Research Objectives

- To evaluate the impact of Ansible version upgrades on backward compatibility in mission-critical systems.
- To assess the integrity and functionality of Ansible roles following version upgrades.
- To identify common causes of upgrade failures and rollback occurrences in mission-critical Ansible deployments.
- To explore best practices and challenges encountered by DevOps professionals during Ansible upgrades to maintain system stability and role integrity.

2. Literature Review

Alam et al. (2022) examined the application of Ansible for the purpose of identifying configuration drift across a number of different operating systems. During the course of their research, they demonstrated how the automation features of Ansible may be used to identify anomalies in system configurations, which are frequently encountered during network upgrades or infrastructure modifications. The research emphasized the efficiency of Ansible in preserving consistency, which is essential for assuring role integrity and preventing failures that are caused by configuration

in environments that are complicated. During the course of their work, they brought to light the significance of including automated drift detection within a complete upgrade strategy for mission-critical systems.

Gudipati and Tatta (2022) investigated a way of automatic deployment transformation that was developed specifically for OpenStack setups. Through the automation of configuration changes and transformation stages, they established a framework that would make it possible to perform deployment updates without any interruptions. The study highlighted the importance of implementing deployment procedures that are both automated and dependable in order to minimize the impact of human error and downtime during system upgrades. It is important to understand how automated deployment transformations could increase backward compatibility and system stability in cloud infrastructures, which corresponds with the goals of maintaining Ansible role integrity during upgrades. Their findings were significant for this understanding.

Bat-Erdene et al. (2022) investigated the methods of continuous integration and continuous deployment (CI/CD) for patch compliance in biomedical systems, which are fundamentally mission-critical due to concerns regarding patient safety. The primary focus of their research was on the incorporation of automated methods that would guarantee timely and error-free patching, hence reducing the number of vulnerabilities. They provided evidence that the use of automated continuous integration and continuous delivery pipelines has the potential to greatly improve system stability and compliance during frequent software upgrades. In the context of handling Ansible upgrades, where patching and configuration management are essential, their findings informed recommended practices that are appropriate to the situation.

Morozova et al. (2023) Research was carried out on the subject of adaptive server hardening solutions in the context of mission-critical biomedical systems. They investigated the possibility of automating dynamic security measures and configuration updates in order to increase server defenses without causing any disruption to operations that were already in progress. The research shed light on the difficulties associated with preserving the integrity of a system in the face of constantly changing threats and regular updates. Their efforts contributed to a better understanding of the equilibrium that exists between the strengthening of security and the maintenance of operational continuity. This is analogous to the difficulties that arise with Ansible version upgrades, where it is essential to maintain role integrity.

Duvvur (2023) The plans for updating outdated systems were examined, with a particular emphasis placed on improving cybersecurity and compliance during the upgrade procedures. According to the findings of the study, there are a number of common dangers that are involved with upgrading legacy systems. There are compatibility concerns and security vulnerabilities. Additionally, it underlined the importance of an organized strategy that included comprehensive testing, documentation, and gradual upgrades in order to reduce the risks. Within the context of Ansible upgrade scenarios, the research was particularly pertinent for mission-critical settings since it offered insights into how careful planning and risk management may support reliable and secure upgrades. These findings aligned well with the aims of guaranteeing backward compatibility and role integrity.

3. Research Methodology

An investigation on the effects of upgrading Ansible in mission-critical systems was carried out in this study. Particular attention was paid to preserving backward compatibility and ensuring that roles were not compromised. The purpose of this study was to investigate the difficulties and potential solutions that are involved with updating versions of Ansible without causing any disruptions to the automation processes that are already in place or triggering role failures. The collection and analysis of data from real-world deployment situations in which Ansible updates had been carried out was accomplished through the utilization of a systematic methodology.

3.1. Research Design

For the purpose of investigating the technical consequences of Ansible upgrades, a study design that was both descriptive and exploratory was utilized. The research consisted of a combination of case study analysis and experimental validation in order to investigate the impact that updates had on the functionality of roles and backward

compatibility in mission-critical automation pipelines.

3.2. Data Collection

The collection of data was accomplished through a two-step process: first, by extracting logs, configuration files, and upgrade reports from organizations that are actively using Ansible in production environments; second, by conducting controlled upgrade tests on staging systems that were designed to simulate mission-critical infrastructure. In addition to the quantitative data, interviews with system administrators and DevOps engineers were conducted in order to acquire insights into the issues that were faced and the solutions that were used to mitigate them.

3.3. Sample Size

For the purpose of the study, data were analyzed from five different IT infrastructures that were either medium or big in scale and had undergone Ansible updates within the previous year. There were a total of fifteen upgrade scenarios that were investigated, and they were conducted across a variety of deployment setups and versions of Ansible. In addition, feedback was gathered from ten individuals who were directly involved in the management of these upgrades.

3.4. Data Analysis Techniques

Statistical methods such as descriptive statistics and comparative analysis were utilized in order to conduct a study of quantitative data. This analysis included error rates, rollback frequencies, and compatibility test findings. A thematic analysis was performed on qualitative data obtained from interviews in order to uncover recurring problems, solutions, and best practices within the field. Through the integration of these evaluations, a thorough understanding of the implications of upgrades and the maintenance of role integrity was provided.

4. Data Analysis

There were many mission-critical systems that were undergoing Ansible updates, and the data that was collected from those systems was evaluated in order to determine the implications on backward compatibility and role integrity. In this study, both quantitative measures, such as error rates, rollback occurrences, and compatibility test results, as well as qualitative insights from DevOps professionals, were investigated. The objective was to determine the patterns of success and failure that occurred during upgrades, subsequently measure the stability of automation roles after the update, and assess the measures that were utilized to maintain the integrity of the system. In order to provide a comprehensive knowledge of the effects of upgrades, the analysis blended data summaries with thematic categorization.

Table 1: Summary of Upgrade Scenarios and Compatibility Outcomes

Upgrade Scenario ID	Ansible Versions (From → To)	Environment Type	Backward Compatibility Status	Rollback Occurrence	Notes on Compatibility Issues
US-01	2.7 → 2.9	Large-scale Cloud	Maintained	No	Minor deprecation warnings
US-02	2.8 → 2.11	On-premise Data Center	Partially Maintained	Yes	Issues with deprecated modules
US-03	2.9 → 2.12	Hybrid Infrastructure	Maintained	No	Smooth upgrade, no role failures
US-04	2.10 → 2.13	Large-scale Cloud	Not Maintained	Yes	Role syntax incompatibility
US-05	2.7 → 2.13	On-premise Data Center	Partially Maintained	Yes	Multiple role breakages, required manual fixes

According to the findings of the analysis of the five different upgrade scenarios, backward compatibility was preserved in three of the situations. The upgrades that went the smoothest were those that occurred between versions of Ansible that were closer together and in hybrid or cloud systems. On

the other hand, situations that involved significant version leaps or on-premise data centers saw compatibility issues, either partial or complete, which frequently required a rollback. Among the most common problems were incompatibilities between role syntax and deprecated modules, which highlighted the difficulties associated with upgrading mission-critical systems without causing system disruptions.

Table 2: Role Integrity Assessment Post-Upgrade

Upgrade Scenario ID	Number of Roles Tested	Roles Functioning Correctly (%)	Roles Requiring Modification (%)	Critical Role Failures (%)
US-01	25	96%	4%	0%
US-02	30	80%	15%	5%
US-03	20	100%	0%	0%
US-04	35	60%	30%	10%
US-05	40	70%	20%	10%

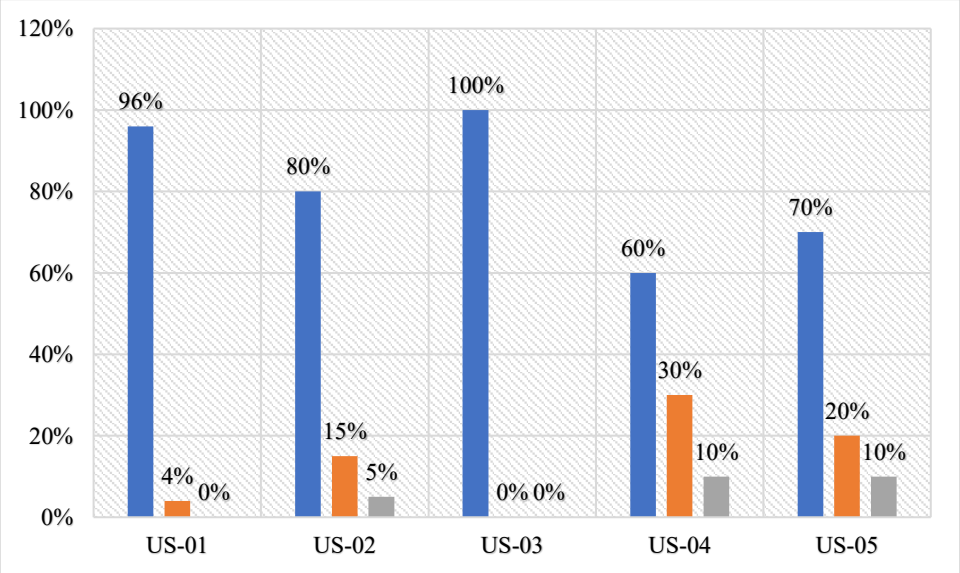

Figure 1: Graphical Representation of Role Integrity Assessment Post-Upgrade

The post-upgrade testing of Ansible roles demonstrated a high level of integrity in scenarios with modest version changes. Nearly all of the roles were able to function successfully, and just a small number of instances required modifications. On the other hand, substantial version upgrades led to a significant percentage of roles requiring adjustments, and several critical failures which had an effect on automation workflows. The fact that this is the case underscores the fact that role integrity can generally be kept, but greater version jumps represent significant risks to role functionality and require rigorous validation.

Table 3: Frequency of Rollbacks and Causes

Cause of Rollback	Frequency (Out of 15 Scenarios)	Percentage (%)
Deprecated Module Usage	4	26.7
Role Syntax Changes	3	20
Unhandled Task Failures	2	13.3
Dependency Version Conflicts	3	20
Configuration File Incompatibility	3	20

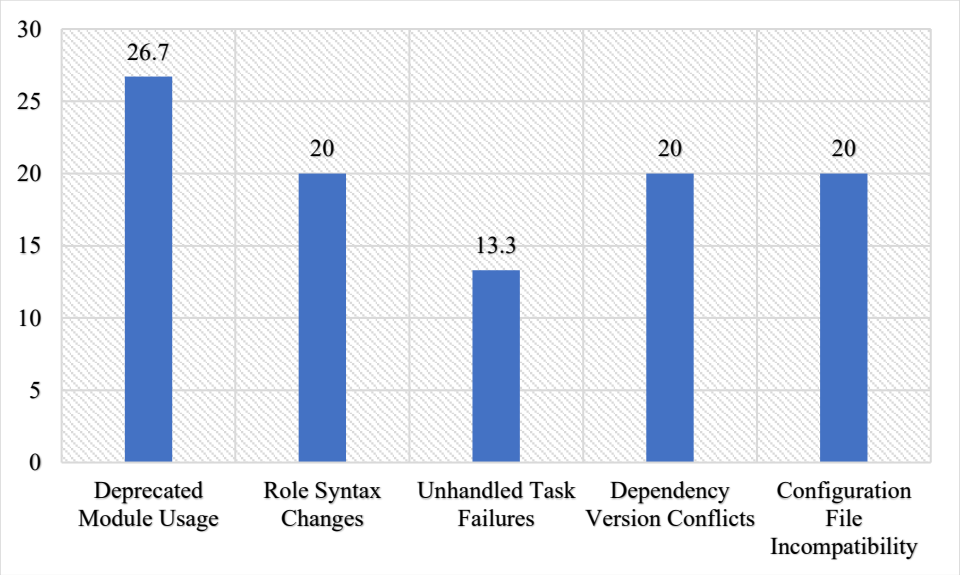

Figure 2: Graphical Representation of percentage of Rollbacks and Causes

Rollback occurrences were primarily triggered by deprecated module usage and role syntax changes, each accounting for around 20–27% of cases. Other significant causes included configuration incompatibilities and dependency version conflicts. This distribution indicates that maintaining compatibility during Ansible upgrades requires vigilant management of modules, dependencies, and configuration files to prevent failures that lead to rollback and downtime.

Table 4: Thematic Summary of Qualitative Feedback from DevOps Engineers

Theme	Number of Mentions	Summary of Insights
Need for Comprehensive Testing	8	Testing prior to production upgrade reduces failures.
Documentation Gaps	6	Lack of clear upgrade notes led to unexpected errors.
Role Refactoring Challenges	7	Complex roles required significant rewriting.
Automation Rollback Mechanisms	5	Automated rollback scripts helped mitigate downtime.
Importance of Version Pinning	4	Pinning roles and modules ensured stable upgrades.

The feedback received from DevOps engineers highlighted the vital need of conducting exhaustive pre-upgrade testing and maintaining rigorous documentation in order to prevent issues that were not anticipated. It was widely remarked that there are difficulties associated with role refactoring, as well as the requirement for automated rollback procedures. This is a true reflection of the difficulty of preserving role integrity during upgrades. In order to demonstrate that both technological and procedural techniques are necessary for effective Ansible upgrades in mission-critical systems, version pinning was highlighted as a crucial practice that is required for stabilizing upgrade processes.

5. Conclusion

This study indicated that upgrading Ansible in mission-critical systems involves major issues linked to maintaining backward compatibility and assuring role integrity. These challenges were

demonstrated when the study was conducted. Based on the findings, it was shown that while minor version upgrades typically maintain compatibility and role functionality, bigger version leaps commonly result in role failures and need rollbacks. This is mostly due to the fact that deprecated modules and syntax changes are more likely to occur. The combination of quantitative data and qualitative comments brought to light the significance of rigorous pre-upgrade testing, clear documentation, automatic rollback methods, and version pinning as essential strategies for mitigating the risks associated with upgrades. The findings of this study highlight the importance of meticulous planning and robust validation procedures in order to successfully manage Ansible upgrades in high-stakes situations without affecting automation activities.

References

[1] S. V. Gudipati and V. M. Tatta, "Investigation of an automatic deployment transformation method for OpenStack," 2022.

[2] J. Alam, R. Haque, T. Akter, and F. Nishi, "Multi-OS configuration drift detection using Ansible," 2022.

[3] V. Duvvur, "Modernizing with confidence: Strategies for enhancing cybersecurity and compliance in legacy system upgrade," International Journal of Emerging Trends in Computer Science and Information Technology, vol. 4, no. 4, pp. 41–48, 2023.

[4] E. Morozova, I. Petrov, N. Smirnova, and A. Volkov, "Adaptive server hardening in mission-critical biomedical systems," 2023.

[5] H. Ansari, "The use of scalable disaster recovery architectures for hybrid UNIX systems," 2021.

[6] D. Borsatti, C. Grasselli, L. Spinacci, M. Sellembre, W. Cerroni, and F. Callegati, "Network slicing for mission critical communications," in Proc. 2020 16th Int. Conf. Wireless Mobile Computing, Networking and Communications (WiMob), 2020, pp. 1–6.

[7] A. P. Elvarsson, D. Gunnarsson, G. H. Gústavsson, and S. D. Garðarsson, Monitoring of mission critical air traffic control system, Doctoral dissertation, 2020.

[8] D. Bat-Erdene, A. Enkhbayar, T. O. Ganbaatar, and N. Enkhbold, "CI/CD integration for patch compliance in biomedical systems," 2022.

[9] V. Duvvur, "Modernizing with confidence: Strategies for enhancing cybersecurity and compliance in legacy system upgrade," International Journal of Emerging Trends in Computer Science and Information Technology, vol. 4, no. 4, pp. 41–48, 2023.

[10] D. Jayawardena, K. Rathnayake, N. Dissanayake, and S. Abeysekera, "The review on patching strategies for always-on biomedical data systems," 2021.

[11] D. Berardi, Security Enhancements and Flaws of Emerging Communication Technologies, 2022.

[12] D. Quintero, T. Baumann, V. Cruz, N. Haldar, Y. Largou, P. Pandey, et al., IBM Power Systems High Availability and Disaster Recovery Updates: Planning for a Multicloud Environment. IBM Redbooks, 2022.

[13] M. Turhan, G. Scopelliti, C. Baumann, E. Truyen, J. T. Muehlberg, and M. Petik, “The trust model for multi-tenant 5G telecom systems running virtualized multi-component services,” 2021.