International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

COMPARATIVE ANALYSIS OF CONFIGURATION MANAGEMENT TOOLS: ANSIBLE VS. SALT STACK

Venkata Soma
New York Mets

Abstract

In the present era of continuous software evolution, configuration management plays a crucial role in automation within the deployment and management of different software systems. This research focuses on the comparative analysis of two well-known configuration management tools such as Ansible and Saltstack. Ansible, which is an open-source IT automation software written in Python, use the "Yet another Markup Language" (YAML) for writing playbooks making it accessible to the developers and the users with minimal programming experience. On the other hand, Saltstack embraces master-minion architecture to foster effective communication. This study investigates different parameters such as scalability, performance and community support which streamlined the findings that provide valuable insights to the IT professionals and system administrators.

Keywords: TDD: Configuration management, Saltstack, Ansible, IT infrastructure, automation, comparative analysis.

I. INTRODUCTION

In the present era of continuous software evaluation, "Configuration management" instruments act as a catalyst in the automation of the management and deployment of various software systems. This research tries to focus on the comparative analysis of the two popular configuration management tools such as "Ansible" and "Saltstack". Ansible is an open-source which is command-line IT automation software application written in Python [1].

a) Project specification

In the contemporary environment of information technology, the requirement for effective, scalable and reliable configuration management tools has become the baseline for software development. The increasing expansion of complicated software systems and the diversified infrastructural environment drive organizations to rely on configuration management tools for automated deployment and maintenance of consistency across their environment [3]. The Saltstack and Ansible stand as the two most useful configuration management tools which pose diverse features and operational efficiency.

b) Aims and objectives

Aim:

This research aims to provide a comprehensive analysis of Ansible and Salt stack by focusing on their scalability, community support and performance.

Objectives:

- To analyse the scalability and performance of both instruments in various environments

- To investigate the community support and the available resource for Saltstack and Ansible
- To provide insights and recommendations for the IT professionals and systems administrators for choosing most appropriate configuration tool

c) Research Questions

RQ 1: How the Ansible and Saltstack comparing in terms of scalability and performance in different IT environments?
RQ 2: what level of community support and resources are available for Ansible and Saltstack?
RQ 3: Which configuration management tool is more suitable for specific use cases in the sports industry?

d) Research Rationale

Ansible is designed as a straightforward, trustworthy and consistent configuration management tool. The configurations of Ansible are descriptions of the fundamental infrastructure data humans can read and process through computers [4]. To manage the system, a secure socket shell is required to create a playbook for the installation of the software on all the nodes. The comparative analysis of the Ansible and Saltstack is essential to understanding their potential strengths and weaknesses within the configuration management process. The insights of this research provide information about the handling process of automation, ease of use and scalability. It assists the sports industry to make more informed decisions in which the tools can meet the evolving requirements and optimise the operational efficiency.

II. LITERATURE REVIEW

a) Research Background

The research background lies within the effectiveness of the two different configuration management tools such s Ansible and Saltstack.
Martyshkin and Biktashev (2022) underline the analysis of computation in cluster configuration management systems by comparing various CCMS tools such as Puppet, Ansible and various others. The main factor that differentiates these CCMS tools are considered as the ease of use, scalability and automation abilities. Among all these tools, Ansible is recognised as an important instrument due to its unique automation feature and simplicity which make it simpler for the smaller clusters [17]. Ansible is considered as the most user-friendly due to its simpler YAML-based syntax and agentless architecture. Ansible is known for its simplicity and ease of use which uses the YAML-based playbooks which makes it ideal for teams with less programming experience. Ansible operated on a push-based model which streamlines the execution of commands from the central server to the targeted machines.
On the other hand, Saltstack offers greater scalability which is crucial for larger environments. It utilised the master-slave architecture with the agents installed on target machines. Compared to the Ansible and Saltstack, it can be stated that Ansible is better for its simplicity, environmental effectiveness and ease of use in which agentless operation is more preferred. Saltstack operates in larger and more complicated environments and requires more scalability and speed.

b) Critical Assessment

In the present scenario of the market, automation is essential for IT management which often becomes complicated and requires comprehensive scaling. Automation simplifies complicated tasks makes developers more manageable and allows them to focus on the value addition to the organisational activities [10]. In this area, Ansible and Saltstack act as a catalyst to drive software development in an effective manner.

Ansible:
Ansible is very simple to set up and use the coding skills which is necessary for the utilisation of the playbook of Ansible. Through the incorporation of Ansible, the users can baler to manage the infrastructure through the utilisation of the coding and promotion of best practices such as code review, version control and continuous integration continuous deployment [11]. The utilisation of agentless architecture and simplicity make Ansible more accessible to network engineers who lack effective programming experience [16].

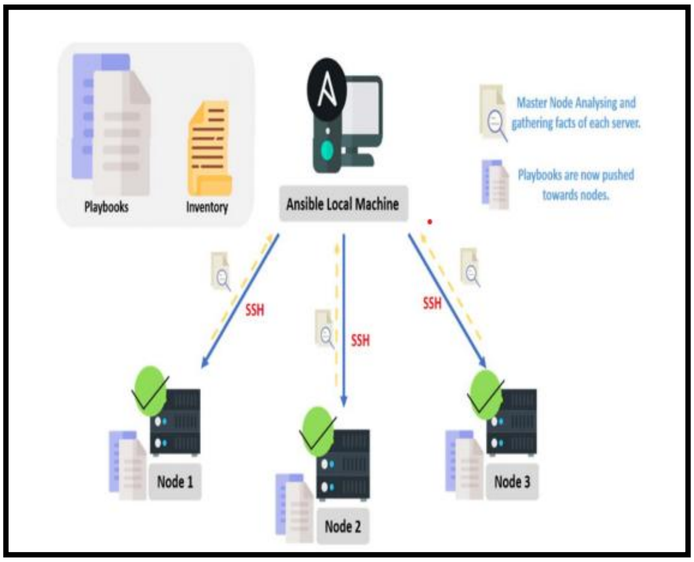
Playbooks
Inventory
Ansible Local Machine
Master Node Analysing and gathering facts of each server.
Playbooks are now pushed towards nodes.
SSH
SSH
SSH
Node 1
Node 2
Node 3

Figure 1: Operational architecture of Ansible [18]

The architecture of Ansible uses the inventory which is a list of nodes comprising IP addresses, servers and databases which is required to be managed. The APIs work as a transparent mode for both private and public cloud services. The playbooks consist of the written code which written as the YAML format which underlines the tasks and executes them through Ansible. The performance of Ansible lies within its agentless nature which leads to the slower execution of times within the larger environment compared to Saltstack which utilises the agents for the faster task execution [15].

Saltstack:
Saltstack is a distributed execution system used for the execution of commands and data queries on remote nodes. This configuration management tool offers a variety of management techniques for IT infrastructure management. It ensures that the configurations are consistent and enforced across all the managed systems through automatic correction of deviations [13].

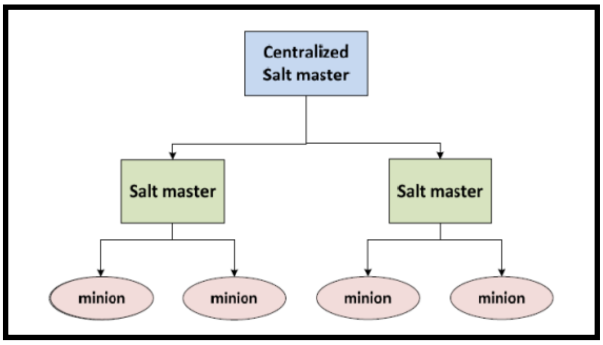

Figure 2: Operational architecture of Saltstack [19]

Saltstack recognised as an open-source, Python-based configuration management system which enables "Infrastructure as Code" for dynamic cloud deployment and management. Its architecture includes a Centralized Salt Stack Master which assist in issuing commands and managing communication. The Salt Stack Master helps to delegate the task to minions, and the Salt Stack Minions assist in the execution of the commands and reporting of outcomes.

c) Linkage to aim

The Ansible allows the sports industries to organise the takes and playbooks into reusable elements. Ansible comprises various components including servers, networking, software and storage. The primary aim of this tool is to maintain the predetermined states of these systems which include the description of the desired state for the system and the utilisation of automation software to ensure the targets remain in this state.

d) Encapsulation of Applications

This tool can used to define and enforce the optimal compliant software states throughout the entire environment. The automatic software installation and the update of the software packages on multiple machines by ensuring the systems are running through the latest versions. It allows the version control of the configurations making it easy to track the changes and maintain the history of the configuration alteration. Through supporting the infrastructure as a code approach, Saltstack enables the users to manage the infrastructure through the utilisation of accurate code. This leverages the automation technique to enable the team to manage any scale ranging from 100 nodes to 1,00,000 nodes [14]. This configuration management tool provides a graphical user interface which is easily usable and user-friendly. This ensures that the network engineers in the sports industry have access to the resources which are essential for the fulfilment of their allotted work responsibilities.

Both Ansible and Saltstack pose significant limitations that hinder their suitability based on their nature and environmental settings. In the case of Ansible, it is more task-oriented which means that it cannot be able to track the desired level of system over the time. The push-bed model of Ansible can be slower sometimes, especially in the large-scale environment. The error handling abilities of Ansible are limited which make it more difficult to handle failures effectively during the execution period.

On the other hand, the pull-based model of Saltstack requires the agents on all the managed nodes which raises complications in the installation process. The flexibility and resource intensity of Saltstack pose significant complications that also make it difficult to consider.

e) Theoretical Framework

Ansible is agentless which underlines that no additional software is required to be installed on the management nodes. Ansible modules for the “NetApp® ONTAP®” software and “Element®

Operation System" that enhance the automation process within the infrastructure. Through using these modules, NetApp customers simplify the complicated initial deployments and automate tasks such as simple volume creation [12].

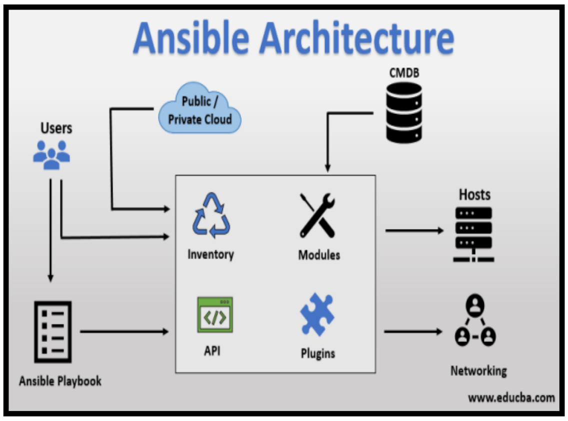

Figure 3: Ansible Architecture [5]

The Ansible configuration management tool uses the declarative approach in which the users can interpret the desired state of the current system in a human-readable and understandable, yet another markup language file called playbooks [6]. This management tool then ensures that the systems are configured to match these descriptions.
The Saltstack configuration management tool delivers vigorous remote executive abilities which enable the developers to run commands in various machines parallelly. It also offers orchestration features which assist in the automation of the complicated IT workflow across diverse environments.

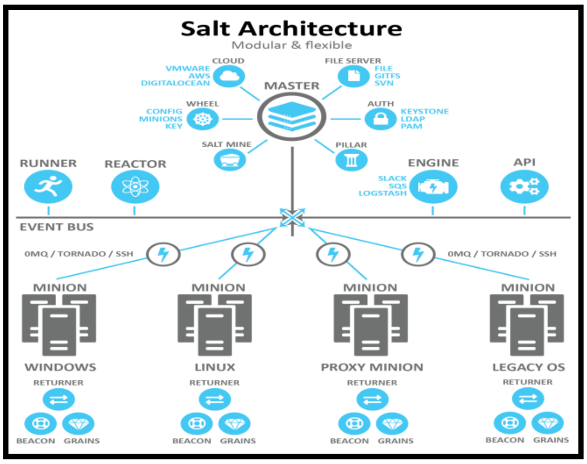

Figure 4: Saltstack Architecture [8]

Saltstack is designed for managing a huge number of servers at a time through the execution of commands across all the servers. It allows the developers to define the desired state of the infrastructural development and automatic enforcement [7].
The event system of Saltstack allows real-time monitoring along with the responses which enables the automatic reactions of the specific events within the infrastructure. The Saltstack assist in the

maintenance of security standards and compliance through the automation and enforcement of security policies. Saltstack supports the management procedure across the cloud and hybrid environments which allows the developers and the users to apply reliable and consistent configurations and automation across the cloud as well as containerised systems [9].

f) Literature gap

The configuration problems lead to various issues such as problems in application performance, security vulnerabilities and the violation of policy complaints. However, selecting the most appropriate tool for the specific use cases is a significant challenge for many organisations, especially sports industries due to the lack of comparative and comprehensive analysis. The mitigation of the configuration management issues heavily relies on the configuration management tools, especially Ansible and Saltstack.

III. METHODOLOGY

a) Research Philosophy

This project is based on comparative analysis of configuration management tools such as Ansible and Saltstack uses the philosophy of interpretivism to emphasize on the perspectives of researchers on this topic. The incorporation of the interpretivism research philosophy within this research assists in the analysis of data related to human activities in response to use of various configuration tools. This philosophy assists in the exploration of human behaviour in response to the use of Ansible and Saltstack and in recognising the suitable features of the most compatible configuration management tool within the evolving landscape of technology.

b) Research approach

The research regarding the assessment of two different configuration tool such as Ansible and Saltstack includes the deductive approach to investigate the efficacy of these two tools. Through using the deductive method this project will provide the opinion of previously working individuals through data collection and analysis. The use of the deductive research approach assists the researchers in the generalization of their ideas and then testing the ideas through the incorporation of specific observations.

c) Research design

This research uses a qualitative method for the analysis of the suitability of Ansible and Saltstack as the most appropriate configuration tools. It will provide a comprehensive analysis of these two configuration tools within the digital environment. This research design assists in the analysis of the scenario-based methodology which highlights the different aspects of Ansible and Saltstack to provide fruitful answers to the research questions.

d) Data collection methods

In this research study, the data collection is conducted through the peer review of the previously published articles accessed through various websites. The gathered information will be analyzed using thematic analysis. The peer-review process assists in ensuring that the used articles and journals provide actionable, accountable and streamlined contributions to the comparative analysis of two different configuration management tools.

e) Ethical consideration

In this research study, the maintenance of the ethical perspectives is crucial. The privacy and permission laws have to be followed when using the confidential information about sports events.

The performances of the Ansible and Saltstack have to be managed through the elimination of the potential biases. Containerization requires appropriate management during the encapsulation in a single package.

IV. RESULTS

a) Critical Analysis

As an open-sourced configuration management tool, Ansible made a substantial impact on global configuration management practices. The agentless architecture and the ease of use foster its adoption across diverse industry settings including healthcare, finance and technology. The YAML-based playbook of Ansible allows users to write and read seamlessly which enables those with minimal coding experience to interpret and manage configurations. The orchestration process of Ansible allows organisations to streamline their operations and reduce the errors posed by humans.

b) Findings and Discussion

Theme 1: scalability and performance of both instruments in various environments

The vigorous documentation aids a significant number of users. Integration with the community supports in larger repository of modules and plugins which further enhances the functionalities of Ansible. The unchangeable nature of Ansible ensures that the configuration is applied consistently through the reduction of risk in configuration drift and the enhancement of system security. Sports organisations have complicated IT environments which support various applications, services and websites. It promotes the interconnectivity between the operation team and IT through its human-readable playbooks. Saltstack is also a powerful open-sourcing orchestration tool and configuration management process. This configuration management tool offers a vigorous solution for the sports industry.

Theme 2: the community support and the available resources for Saltstack and Ansible

The automation capabilities of Ansible allow the streamlined management of the sports environment which reduces the time and effort required for routine activities. Ansible assists in maintaining the security of the IT system through the automation of the application for the security patches. The event-driven architecture of Saltstack allows the rapid execution of the allotted tasks. The use of the ZeroMQ message process fosters high-speed communication between the minions and the master. It enables the real-time alteration and monitoring of configuration activities. The modular design of Saltstack allows extensive customisation through which users can be able to read and write the custom modules to meet specific requirements.

Theme 3: insights and recommendations for the IT professionals and systems administrators for choosing the most appropriate configuration tool

The findings of this research can provide valuable insights to IT professionals, decision-makers and system administrators in choosing the most stable configuration management tool that is tailored to the specific requirements of the ongoing market structure. The ability of this configuration management tool to interact with the wider ecosystem expedites the holistic automation solutions. This further supports the entire IT lifecycle from provisioning of decomposition. Although the community of Saltstack is smaller compared with Ansible, the incorporation of VMware has strengthened the development process driven by Saltstack.

c) Evaluation

The event-driven architecture of this configuration tool allows real-time configuration monitoring and alteration. This ability is beneficial for sports organisations which are required to ensure that the IT systems are always secure and up-to-date, particularly during live events. The sports industries often face higher levels of traffic volumes during major events. The high-speed communication and ability to instantly execute the task drive the intricate performance and flexibility during peak times.

V. CONCLUSION

From the above analysis, it can be concluded that the comparative analysis between Saltstack and Ansible underlines the activities of each configuration tool which provides a holistic understanding for the IT professionals. The simplicity, agentless architecture and ease of use make Ansible an attractive choice for organisations seeking engaging and straightforward automation solutions. On the contrary, the master-minion architecture along with the event-driven abilities offers vigorous scalability and performance for larger and more complicated environments. The ability of Saltstack for the execution of commands and management of configuration in real-time provides opportunities for the IT professionals associated with the sports industry. In conclusion, the choice between Ansible and Saltstack has to be guided through the specific operational requirements along with the organisational complications.

VI. RESEARCH RECOMMENDATIONS

Future research can expand on this research study through the investigation of the performance evaluation of these two configuration management tools along with the exploration of other configuration management tools such as Terraform, Puppet and Chef. The enhancement of the effectiveness and credibility of these tools through best practices such as utilisation of version control, maintenance of simplicity and customisation of outputs. It further supports the sports industry in achieving reliable and effective automation solutions.

VII. FUTURE WORK

Future studies can explore the potential advantages of emerging configuration management tools in the performance evaluation of the sports industry. The enhancement of Ansible's credibility is rooted in the maintenance of simplicity in the configuration. Through using version control and keeping playbooks, inventories, roles and variables files and another version control system, this tool assists in auditing in the time of alteration of rules and regulations. On the other hand, it is necessary to avoid complications by using straightforward modules to maintain the simplicity of the Saltstake configuration. The simplified configuration assists in streamlined configuration management as it is easier to maintain and understand. Clear and transparent outcomes can make it easier to understand the execution and output of the current states by increasing the clarity and trustworthiness of the automation process.

REFERENCES

1. J. Freeman and J. Keating, "Mastering Ansible: Automate configuration management and overcome deployment challenges with Ansible," Packt Publishing Ltd., 2021. Available at:https://books.google.com/books?hl=en&lr=&id=ec5KEAAAQBAJ&oi=fnd&pg=PP1&dq=Ansible+configuration+management+tool&ots=So-hQ8MIEl&sig=8EGtoqQolKByvdXmU-A9h3p-fBY
2. M. Wurster, U. Breitenbücher, M. Falkenthal, C. Krieger, F. Leymann, K. Saatkamp, and J. Soldani, "The essential deployment metamodel: a systematic review of deployment automation technologies," SICS Software-Intensive Cyber-Physical Systems, vol. 35, pp. 63-75, 2020. Available at:https://link.springer.com/article/10.1007/s00450-019-00412-x
3. J. Nørbjerg and Y. Dittrich, "The Never-Ending Story–How companies transition to and sustain continuous software engineering practices," Journal of Systems and Software, vol. 213, p. 112056, 2024. Available at:https://doi.org/10.1016/j.jss.2024.112056
4. R. Kostromin, "Survey of software configuration management tools of nodes in heterogeneous distributed computing environment," In ICCS-DE, pp. 156-165, 2020. Available at:https://ceur-ws.org/Vol-2638/paper15.pdf
5. Educba.com, "Ansible Architecture," Available at:https://www.educba.com/ansible-architecture/, (accessed on: 30 July, 2024)
6. B. Choi and E. Medina, "Understanding Data Types and File Formats for Ansible," In Introduction to Ansible Network Automation: A Practical Primer, pp. 211-257, Berkeley, CA: Apress, 2023. Available at:https://link.springer.com/chapter/10.1007/978-1-4842-9624-0_5
7. O. Tomarchio, D. Calcaterra, and G.D. Modica, "Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks," Journal of Cloud Computing, vol. 9, no. 1, p. 49, 2020. Available at:https://link.springer.com/article/10.1186/s13677-020-00194-7
8. Saltproject.io, "Salt system architecture," Available at:https://docs.saltproject.io/en/3006.0/topics/salt_system_architecture.html, (accessed on: 30 July 2024)
9. M. Valkeinen, "Cloud Infrastructure Tools For Cloud Applications: Infrastructure management of multiple cloud platforms," (Master's thesis), 2022. Available at:https://trepo.tuni.fi/handle/10024/137424
10. D. Kedziora, A. Leivonen, W. Piotrowicz, and A. Öörni, "Robotic process automation (RPA) implementation drivers: Evidence of selected Nordic companies," Issues in Information Systems, vol. 22, no. 2, pp. 21-40, 2021. Available at:https://harisportal.hanken.fi/sv/publications/robotic-process-automation-rpa-implementation-drivers-evidence-of
11. M. Gupta, M.N. Chowdary, S. Bussa, and C.K. Chowdary, "Deploying Hadoop architecture using ansible and terraform," In 2021 5th International Conference on Information Systems and Computer Networks (ISCON), pp. 1-6, IEEE, 2021. 10.1109/ISCON52037.2021.9702299
12. B. Bratus, "Orchestration of network applications using middleware for 5G enhanced autonomous robots," 2023. Available at:https://uobrep.openrepository.com/handle/10547/626264
13. X. Sun, W. Luo, J.T. Gu, A. Ganesan, R. Alagappan, M. Gasch, L. Suresh, and T. Xu, "Automatic reliability testing for cluster management controllers," In 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), pp. 143-159, 2022. Available at:https://www.usenix.org/conference/osdi22/presentation/sun

14. M. El Rajab, L. Yang, and A. Shami, "Zero-touch networks: Towards next-generation network automation," Computer Networks, vol. 243, p. 110294, 2024. Available at:https://doi.org/10.1016/j.comnet.2024.110294

15. S. Wågbrant and V. DahlénRadic, "Automated Network Configuration: A Comparison Between Ansible, Puppet, and SaltStack for Network Configuration," 2022. Available:https://www.diva-portal.org/smash/record.jsf?pid=diva2:1667034.

16. B. Choi and E. Medina, "Is Ansible Good for Network Automation?" in Introduction to Ansible Network Automation: A Practical Primer, Berkeley, CA: Apress, 2023, pp. 3-30. Available:https://link.springer.com/chapter/10.1007/978-1-4842-9624-0_1.

17. A. I. Martyshkin and R. A. Biktashev, "Research and Analysis of Computing Cluster Configuration Management Systems," in International Russian Automation Conference, Cham: Springer International Publishing, Sep. 2022, pp. 194-205. Available:https://link.springer.com/chapter/10.1007/978-3-031-22311-2_20.

18. S. Likitha, "Automation of Server Configuration Using Ansible," Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET), vol. 10, no. 6, pp. 4109-4113, 2022. Available:https://www.academia.edu/download/89100391/Automation_of_Server_Configuration_Using_Ansible.pdf.

19. "OpenStack Post Deployment Config, Logs and Metrics Management System," Ajetijournal.com, 2018. Available:https://www.ajetijournal.com/open-access/openstack-post-deployment-config-logs-and-metrics-management-system-267.pdf. (Accessed on: 16.08.2024).