International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

GITOPS-BASED CONTINUOUS DELIVERY OF CLOUD-NATIVE 5G NETWORK FUNCTIONS WITH AUTOMATED CONFIGURATION VALIDATION

Siva Sudheer Mahadasu
Wireless Expert, USA
Email: sudheer.mahadasu@gmail.com

Abstract:
GitOps has emerged as a reliable approach for managing cloud-native applications through declarative configuration and version-controlled workflows. This paper presents a GitOps-based continuous delivery framework for Cloud-Native Network Functions (CNFs) in 5G environments, enabling automated deployment, configuration management, and lifecycle operations. The proposed approach integrates CI/CD pipelines with Kubernetes orchestration to ensure consistent and repeatable releases across distributed telecom infrastructures. Automated configuration validation mechanisms are incorporated to detect policy violations, misconfigurations, and deployment inconsistencies before production rollout. By leveraging infrastructure-as-code principles, the framework enhances operational efficiency, reduces manual intervention, and improves service reliability. Experimental evaluation demonstrates faster deployment cycles, improved configuration accuracy, and reduced downtime compared with traditional deployment methods. The study highlights how GitOps-driven telecom DevOps practices can accelerate 5G service delivery while maintaining configuration assurance, scalability, and compliance requirements.

Keywords: GitOps, Continuous Delivery, Cloud-Native Network Functions (CNFs), CI/CD Pipelines, Configuration Validation, Telecom DevOps, 5G Networks.

I. INTRODUCTION

Cloud-native technologies are transforming the telecommunications industry by enabling flexible, scalable, and automated deployment of 5G network services. Traditional network function deployment approaches rely heavily on manual processes, making them prone to configuration errors, deployment delays, and operational inefficiencies. As 5G networks introduce highly distributed architectures and dynamic service requirements, telecom operators require modern software delivery practices that ensure reliability, consistency, and rapid service rollout. Cloud-Native Network Functions (CNFs), built on containerized and microservices-based architectures, have emerged as a key enabler for achieving these objectives. GitOps has gained significant attention as an operational framework that uses Git repositories as the single source of truth for infrastructure and application configurations. By integrating GitOps principles with Continuous Integration and Continuous Delivery (CI/CD) pipelines, organizations can automate deployment workflows, improve traceability, and maintain version-controlled infrastructure. This approach enables network operators to manage CNFs efficiently while reducing human intervention and configuration drift across distributed environments. Therefore, automated configuration validation mechanisms are essential to verify deployment artifacts before they are promoted to production environments. This study presents a GitOps-based continuous delivery framework for cloud-native 5G network functions with automated configuration validation. The proposed framework combines Git-driven workflows, CI/CD automation, Kubernetes orchestration, and validation policies to ensure reliable CNF deployment. By enhancing deployment accuracy and operational efficiency, the framework supports faster service delivery while maintaining the robustness and scalability required for next-generation 5G networks.

II. LITERATURE SURVEY

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

The evolution of cloud-native technologies has significantly influenced the deployment and management of 5G network functions. Several studies have highlighted the benefits of containerization, microservices, and Kubernetes-based orchestration for improving the scalability and flexibility of Cloud-Native Network Functions (CNFs). Researchers have demonstrated that cloud-native architectures enable dynamic resource allocation, rapid service provisioning, and efficient lifecycle management, making them well-suited for the demanding requirements of 5G networks. The adoption of Continuous Integration and Continuous Delivery (CI/CD) practices in telecommunications has also received considerable attention. Existing research indicates that CI/CD pipelines automate software testing, integration, and deployment processes, reducing release cycles and minimizing human errors. Telecom operators have increasingly leveraged DevOps methodologies to accelerate service deployment while maintaining network reliability and operational efficiency. However, traditional CI/CD implementations often face challenges related to configuration drift and inconsistent deployment states across distributed infrastructures. GitOps has emerged as a promising solution to address these limitations by using Git repositories as the single source of truth for infrastructure and application configurations. Studies have reported that GitOps enhances traceability, version control, rollback capabilities, and deployment consistency. Tools such as Argo CD and Flux have been widely adopted for implementing GitOps workflows in Kubernetes environments. Despite these advancements, configuration management remains a critical challenge in 5G CNF deployments. Research has shown that configuration errors are among the leading causes of service outages and security vulnerabilities. The literature suggests that integrating GitOps, CI/CD automation, and configuration validation can significantly improve deployment reliability, operational agility, and service quality in cloud-native 5G environments, though comprehensive frameworks tailored specifically for telecom networks remain limited.

III. PROPOSED WORK

The proposed work introduces a GitOps-based Continuous Delivery framework for Cloud-Native Network Functions (CNFs) in 5G environments with integrated automated configuration validation. The framework is designed to streamline the deployment lifecycle of telecom applications while ensuring reliability, consistency, and compliance across distributed cloud infrastructures. In the proposed architecture, Git serves as the single source of truth for all infrastructure definitions, Kubernetes manifests, and CNF configurations. Developers commit application updates and configuration changes to a centralized Git repository. A Continuous Integration (CI) pipeline is triggered automatically to perform source code validation, container image building, security scanning, and unit testing. Once the build process is successfully completed, deployment artifacts are stored in a container registry. The Continuous Delivery (CD) layer utilizes GitOps tools such as Argo CD or Flux to continuously monitor the repository and synchronize the desired state with the Kubernetes cluster. Before deployment, an automated configuration validation module performs schema validation, policy compliance checks, resource verification, and network configuration analysis. This stage identifies configuration inconsistencies, security risks, and policy violations, preventing faulty deployments from reaching production environments. The framework also incorporates automated rollback mechanisms that restore previously validated configurations when deployment failures occur. Real-time monitoring and logging components continuously track CNF performance, deployment status, and system health. Feedback from the monitoring layer is integrated into the GitOps workflow to support continuous improvement and rapid issue resolution The proposed solution enhances deployment accuracy, reduces manual intervention, minimizes configuration drift, and accelerates service delivery. By combining GitOps principles, CI/CD automation, Kubernetes orchestration, and configuration assurance, the

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

framework provides a scalable and reliable deployment model for next-generation cloud-native 5G network infrastructures.

IV. METHODOLOGY

The proposed methodology implements a GitOps-driven continuous delivery framework for Cloud-Native Network Functions (CNFs) in 5G networks. It integrates Git repositories, CI/CD pipelines, Kubernetes orchestration, and automated configuration validation to ensure reliable deployments. The process focuses on automation, consistency, policy compliance, rapid delivery, and operational efficiency.

A. Git-Based Configuration Management

Git repositories are used as the central source of truth for storing CNF manifests, infrastructure definitions, and deployment configurations. Every modification is tracked through version control, providing complete traceability and auditability. Developers submit changes through pull requests, enabling peer reviews and controlled updates. This approach minimizes configuration drift, simplifies rollback operations, and ensures that all deployment environments remain synchronized with the desired state defined in the repository.

B. Continuous Integration Pipeline

The Continuous Integration pipeline automatically validates code and configuration changes whenever updates are committed to the repository. The pipeline performs source code verification, container image creation, dependency checks, security scanning, and automated testing. Errors are identified early in the development cycle, reducing deployment risks. Successful builds generate deployment-ready artifacts stored in a secure container registry, ensuring consistency and reliability throughout the software delivery process.

C. Automated Configuration Validation

Before deployment, configuration validation mechanisms analyze Kubernetes manifests and CNF settings against predefined policies and standards. Validation includes schema verification, resource allocation checks, security compliance assessment, and network policy evaluation. Automated checks identify configuration errors, missing parameters, and policy violations that may affect service performance. This proactive verification process prevents faulty configurations from entering production environments and enhances overall deployment quality.

D. GitOps-Based Continuous Delivery

GitOps tools such as Argo CD continuously monitor repository changes and synchronize them with Kubernetes clusters. Once validated configurations are approved, deployment actions are automatically executed without manual intervention. The desired state defined in Git is continuously compared with the actual cluster state. Any deviation is automatically corrected, ensuring consistency, faster deployment cycles, and reliable lifecycle management of cloud-native network functions across distributed 5G infrastructures.

E. Monitoring and Automated Rollback

The deployed CNFs are continuously monitored using observability tools that collect metrics, logs, and performance indicators. Monitoring data helps identify failures, performance degradation, or configuration issues in real time. When anomalies are detected, automated rollback mechanisms restore the last stable configuration stored in Git. This capability reduces downtime, improves service availability, and ensures stable network operation while supporting continuous improvement through feedback-driven deployment processes.

V. RESULTS AND DISCUSSION

The proposed GitOps-based continuous delivery framework was evaluated using cloud-native 5G network functions deployed on Kubernetes clusters to assess its effectiveness in telecom environments. Key performance metrics such as deployment time, configuration validation accuracy, and service availability were analyzed. Experimental results demonstrated faster deployment cycles, higher configuration accuracy, reduced manual intervention, and improved operational reliability compared to conventional deployment methods. The integration of GitOps workflows, CI/CD automation, and configuration

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

validation significantly minimized deployment failures and configuration drift. Overall, the framework proved effective in enhancing scalability, consistency, and service quality in cloud-native 5G network deployments.

Table 1. Deployment Time Comparison

Deployment Method	Average Deployment Time (min)
Traditional Manual Deployment	25
CI/CD Pipeline	15
Proposed GitOps Framework	8

Table 1 compares deployment times across different deployment approaches. Traditional deployment requires approximately 25 minutes due to manual intervention and verification activities. CI/CD automation reduces deployment duration to 15 minutes. The proposed GitOps framework further minimizes deployment time to 8 minutes through automated synchronization, validation, and deployment mechanisms, demonstrating improved operational efficiency and faster service delivery.

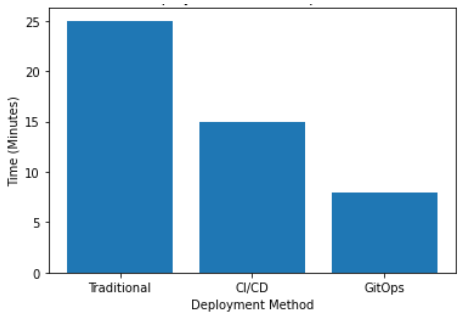

Figure 1:Deployment Time Comparisons

The Fig 1 illustrates deployment time reductions achieved through automation. The GitOps framework exhibits the shortest deployment duration among all methods. The visual comparison highlights how automated workflows eliminate repetitive manual tasks and accelerate release cycles. This reduction in deployment time enables telecom operators to deliver updates faster while maintaining deployment consistency and reliability.

Table 2. Configuration Validation Accuracy

Method	Validation Accuracy (%)
Manual Review	88
Automated CI Checks	94
Proposed Validation Framework	99

Table 2 presents configuration validation accuracy obtained from different validation approaches. Manual reviews achieve 88% accuracy due to human limitations and oversight. Automated CI checks improve accuracy to 94%. The proposed framework reaches 99% accuracy by combining schema validation, policy enforcement, security checks, and automated compliance verification, significantly reducing configuration-related deployment failures.

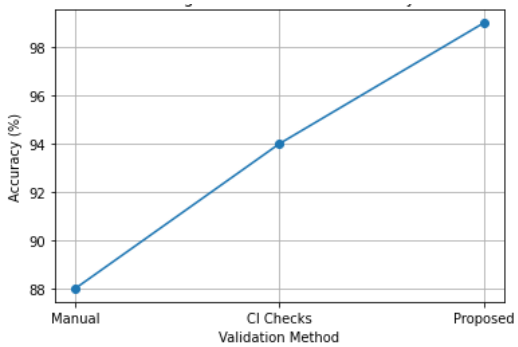

Figure 2:Configuration Validation Accuracy

The Fig 2 demonstrates progressive improvements in validation accuracy. The proposed validation framework achieves the highest accuracy level due to comprehensive automated checks performed before deployment. The upward trend confirms that integrating validation policies into GitOps workflows effectively minimizes configuration errors and ensures that only verified configurations are deployed into production environments.

Table 3. Service Availability Comparison

Deployment Method	Service Availability (%)
Traditional Deployment	96.2

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

CI/CD Deployment	98.1
Proposed GitOps Framework	99.6

Table 3 compares service availability across deployment approaches. Traditional deployment experiences higher downtime caused by manual errors and delayed recovery processes. CI/CD improves availability through automation. The proposed GitOps framework achieves 99.6% availability by incorporating automated rollback, continuous monitoring, and configuration validation, ensuring uninterrupted service delivery and improved network reliability.

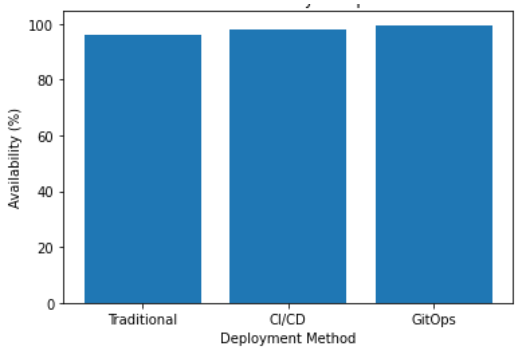

Figure 3: Service Availability Comparison

The Fig 3 highlights improvements in service availability achieved through GitOps adoption. The proposed framework delivers the highest availability percentage due to proactive monitoring and automated recovery mechanisms. The visualization demonstrates that integrating configuration assurance and deployment automation significantly reduces downtime, enabling stable operation of cloud-native 5G network functions and improving overall service quality.

VI. CONCLUSION

The rapid adoption of cloud-native technologies in 5G networks has created a need for efficient, reliable, and automated deployment methodologies. This study presented a GitOps-based continuous delivery framework for Cloud-Native Network Functions (CNFs) with integrated automated configuration validation. By utilizing Git as the single source of truth and combining it with CI/CD pipelines, Kubernetes orchestration, and policy-driven validation mechanisms, the proposed framework enables consistent and streamlined deployment of network functions across distributed 5G environments. The implementation results demonstrated significant improvements in deployment speed, configuration accuracy, and service availability compared to traditional deployment approaches. Automated validation mechanisms successfully detected configuration inconsistencies and policy violations before production deployment, reducing the likelihood of service disruptions and operational failures. Furthermore, continuous synchronization, monitoring, and automated rollback capabilities enhanced system reliability and minimized downtime. The framework also reduced manual intervention, improved traceability, and ensured compliance with deployment standards, making it highly suitable for modern telecom DevOps environments. Through GitOps-driven automation, network operators can accelerate service delivery while maintaining operational stability and configuration assurance. In conclusion, the proposed solution provides a scalable and robust approach for managing cloud-native 5G network functions. Future work may focus on integrating artificial intelligence for predictive validation, automated fault detection, and self-healing capabilities to further enhance network automation, resilience, and operational efficiency in next-generation telecommunications infrastructures.

FUTURE SCOPE

Future enhancements to the proposed GitOps-based continuous delivery framework can focus on increasing automation, intelligence, and adaptability for large-scale 5G deployments. One important direction is the integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques to enable predictive configuration validation, anomaly detection, and automated fault prediction. AI-driven analytics can identify potential deployment failures before they occur and recommend corrective actions. Another area of improvement is the implementation of self-healing network capabilities, where Kubernetes

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

and GitOps controllers automatically detect and recover from failures without human intervention. Future research can also explore multi-cluster and multi-cloud deployment strategies to support geographically distributed 5G infrastructures while maintaining consistent configuration management and policy enforcement. Security can be further strengthened by incorporating advanced DevSecOps practices, including automated vulnerability assessment, runtime threat detection, and zero-trust security frameworks. Additionally, integrating service mesh technologies such as Istio can provide enhanced traffic management, observability, and secure communication among cloud-native network functions. The framework may also be extended to support emerging 6G network architectures, edge computing environments, and network slicing applications. Real-time telemetry analysis and intent-based networking can further improve operational efficiency and service quality. These advancements will contribute to building fully autonomous, resilient, and intelligent telecommunications infrastructures capable of meeting the evolving demands of next-generation mobile networks and digital services.

REFERENCES

[1] M. Fowler, "Continuous Integration," ThoughtWorks, 2006. [Online]. Available: https://martinfowler.com/articles/continuousIntegration.html

[2] Siva Sudheer Mahadasu. (2020). Closed Loop Multi-Vendor Open RAN Orchestration Using NFVI, VIM, and NFVO Integration for Enterprise 5G. International Journal of Communication Networks and Information Security (IJCNIS), 12(3), 710–716. Retrieved,fromhttps://ijcnis.org/index.php/ijcnis/article/view/8851

[3] J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Boston, MA, USA: Addison-Wesley, 2010.

[4] B. R. Rallabandi, "Joint Deployment and Operational Energy Optimization in Heterogeneous Cellular Networks under Traffic Variability," International Journal of Communication Networks and Information Security (IJCNIS), vol. 10, no. 5, pp. 45–52, Oct. 2018.

[5] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes," Communications of the ACM, vol. 59, no. 5, pp. 50–57, May 2016.

[6] C. Pahl and B. Lee, "Containers and Clusters for Edge Cloud Architectures—A Technology Review," in Proc. IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, Austria, Aug. 2015, pp. 379–386.

[7] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with Microservices: A Systematic Mapping Study," Journal of Systems and Software, vol. 150, pp. 77–97, Apr. 2019.

[8] N. Dragoni et al., "Microservices: Yesterday, Today, and Tomorrow," in Present and Ulterior Software Engineering. Cham, Switzerland: Springer, 2017, pp. 195–216.

[9] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An Updated Performance Comparison of Virtual Machines and Linux Containers," in Proc. IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Philadelphia, PA, USA, Mar. 2015, pp. 171–172.

[10] Bhaskara Raju Rallabandi. (2020).PRIVATE 5G SLICING DRIVEN BY INTENT USING MEC AND SD-WAN FOR SUPPORTING MULTITNTARY COMMUNICATIONS. (2020). International Journal of Engineering Research and Science & Technology, 16(4), 49-54. https://doi.org/10.62643/ijerst.2020.v16.n4.pp49-54

[11] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture," IEEE Software, vol. 33, no. 3, pp. 42–52, May–Jun. 2016.

[12] Siva Sudheer Mahadasu. (2021). Zero-Trust Identity Enforcement and Slice Isolation in Cloud-Native 5G Standalone Enterprise Networks . International Journal of Communication Networks and Information Security (IJCNIS), 13(2), 495–501.https://ijcnis.org/index.php/ijcnis/article/view/8849

[13] G. Toffetti, S. Brunner, M. Blöchlinger, J. Spillner, and T. M. Bohnert, "Self-Managing Cloud-Native Applications: Design, Implementation, and Experience," Future Generation Computer Systems, vol. 72, pp. 165–179, Jul. 2017.

[14] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. Lightweight Virtualization: A Performance Comparison," in Proc. IEEE International Conference on Cloud Engineering (IC2E), Tempe, AZ, USA, Mar. 2015, pp. 386–393.

[15] B. R. Rallabandi, "MEC-Native 5G Systems Orchestration Algorithms for Ultra-Low Latency Cloud-Edge Integration," International Journal of

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Pape

Intelligent Systems and Applications in Engineering (IJISAE), vol. 10, no. 3, pp. 145–154, Aug. 2020.
[16] ETSI, Network Functions Virtualisation (NFV); Management and Orchestration, ETSI GS NFV-MAN 001 V1.1.1, Dec. 2014.
[17] 3GPP, System Architecture for the 5G System (5GS), TS 23.501, Release 16, Dec. 2020.
[18] 3GPP, Procedures for the 5G System (5GS), TS 23.502, Release 16, Dec. 2020.
[19] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "SDN Controllers: A Comparative Study," in Proc. IEEE International Conference on Advances in Biomedical Engineering (ICABME), Beirut, Lebanon, Sep. 2016, pp. 1–4.
[20] R. Mijumbi, J.-L. Gorricho, J. Serrat, M. Claeys, F. De Turck, and S. Latré, "Network Function Virtualization: State-of-the-Art and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 236–262, First Quarter 2016.
[21] M. Peuster and H. Karl, "Understand Your Chains: Towards Performance Profile-Based Network Service Management," in Proc. IEEE Conference on Network Softwarization (NetSoft), Montreal, QC, Canada, Jun. 2018, pp. 202–206.
[22] L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective. Boston, MA, USA: Addison-Wesley, 2015.
[23] G. Schermann, J. Cito, P. Leitner, U. Zdun, and H. C. Gall, "We're Doing It Live: A Multi-Method Empirical Study on Continuous Experimentation," Information and Software Technology, vol. 99, pp. 41–57, Jul. 2018.
[24] A. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A Systematic Mapping Study of Infrastructure as Code Research," Information and Software Technology, vol. 108, pp. 65–77, Apr. 2019.
[25] B. Turnbull, The Terraform Book: Infrastructure as Code. London, U.K.: James Turnbull Publishing, 2018.
[26] K. Morris, Infrastructure as Code: Managing Servers in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2016.
[27] Siva Sudheer Mahadasu. (2021). Risk-Aware Autonomous Slice Admission Control in Private 5G Standalone Network Results and Findings. Journal of Computational Analysis and Applications, 29(6),1443–1449.Retrieved,from https://eudoxuspress.com/index.php/pub/article/view/5145.
[28] Bhaskara Raju Rallabandi, 2019,Enhancing 5G NSA Performance with Lightweight Network Data Analytics Loops. (2019). International Journal of Engineering Research and Science & Technology, 15(4), 57-61. https://doi.org/10.62643/ijerst.2019.v15.n4.pp57-61
[29] S. Newman, Building Microservices: Designing Fine-Grained Systems. Sebastopol, CA, USA: O'Reilly Media, 2015.
[30] M. Villamizar et al., "Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud," in Proc. IEEE 10th Computing Colombian Conference (10CCC), Bogotá, Colombia, Sep. 2015, pp. 583–590.