International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 4, Issue 3, 98-106, 2023
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V4I3P111

Original Article

Zero-Downtime CI/CD Production Deployments for Insurance SaaS Using Blue/Green Deployments

Gowtham Reddy Enjam[1], Komal Manohar Tekale[2], Sandeep Channapura Chandragowda[3]
[1,2,3]Independent Researcher, USA.

Abstract - *The continued evolution of the Software-as-a-Service (SaaS) within the insurance industry demands that deployment strategies be installed to guarantee availability, scalability, and fault tolerance and minimized downtime. Traditional deployment approaches (rolling updates/canary releases) are capable of introducing both risk and latency to important applications. A framework that would be effective to apply in order to achieve zero-downtime deployments in Continuous Integration/Continuous Deployment (CI/CD) pipelines would be the blue /green deployment strategy. This paper also elaborates on conceptual underpinning, issue and practical application of Blue/Green deployments in Insurance SaaS production systems. A mix of Blue/Green strategies and CI/CD practices will enable the insurance companies to ensure that their availability is not low and regulatory scrutiny is not compromised and that sensitive customer information is not at the risk. The provided methodology provides end to end architecture like automated testing, monitoring, rollback and orchestration within the cloud. Results of a virtualized SaaS environment reveal reduced Mean Time to Recovery (MTTR), increased deployment success and improved customer experience. The provided paper will contribute to the DevOps realm of insurance SaaS sphere as it will provide the systematic approach, as well as expand on the trade-off that is possible when creating the Blue/Green deployments.*

Keywords *- Blue/Green Deployment, Zero-Downtime, CI/CD, Insurance SaaS, DevOps, Cloud Computing, Continuous Delivery.*

1. Introduction

1.1. Background

Insurance industry is undergoing a radical digital remodeling, following the adoption of cloud-native technologies, automation, and data-driven making of decisions. SaaS insurance software should be able to process a lot of sensitive data including millions of customer records and dynamically evolving policy transactions and complex claims processing. The availability and reliability of the system are very critical due to the sensitivity of such operations because few minutes of servicing can result in the loss of important services, money as well as the confidence of the customers. [1-4] what is even more, the lapses or stalling of such controlled systems can lead to violations of the compliance with such frameworks as the HIPAA, GDPR, and IRDAI, and legal and reputational consequences. To obtain the same effect, i.e. fast innovation without disturbing the stability, the organizations have increasingly turned to Continuous Integration and Continuous Deployment (CI/CD) pipelines. With CI/CD, it becomes possible to release at shorter intervals, automated testing, and as well as deployment that is easy, hence ability to roll out new features and changes becomes faster. However, the faster, more agile CI/CD the pipelines also pose issues with the consistency of production, which is of utmost importance in these mission-critical systems, such as insurance SaaS, where there is no tolerance for downtime or error. This agility vs. stability dilemma leads to emerging deployments practices, e.g. Blue/Green deployments, that facilitate the ongoing upgrades of insurer's platforms, whilst ensuring continuity of service and regulatory compliance.

1.2. Importance of Zero-Downtime CI/CD Production Deployments

- **Business Continuity:** The insurance sector needs the constant digital access to services such as policy issuance, payment of premiums and claims. With zero-downtime deployments, the seamless service delivery to the customers, the agents and the partners through the upgrading of the software is maintained. Such consistency mitigates any disturbance to the operations and upholds trust to the credibility of the platform.
- **Regulatory Compliance:** The field in which insurance companies operating is so regulated, and they are regulated by the laws such as HIPAA, GDPR, and IRDAI. These types of structures are in need of systems that handle sensitive customer information at all times. Zero-downtime rollouts can not only stop compliance failures caused by unexpected downtimes, but also demonstrate the capability to achieve service availability requirements during audit.
- **Customer Experience:** The customer retention and customer satisfaction are directly related to downtime. Even a few minutes of the downtime can reduce the speed of the claims settlement, disrupt the financial transactions, and frustrate policyholders. The zero-downtime CI/CD processes will ensure that value is added to customers on a regular basis and in terms of feature and security enhancements, and this will not happen due to service interruption, and this will create customer loyalty and trust.

- **Operational Efficiency:** Traditional deployments are normally accompanied by scheduled maintenance or manual reactions that leads to deficiency of efficiencies and high cost of running the deployments. The automation of processes and new technology like Blue/Green deployments that reduce human error and the fact that releases are easier to manage facilitate the zero-downtime solutions. It is through this performance that the innovation cycles can be done faster and stability of the system is not lost.
- **Competitive Advantage:** Since the rapid digitalization process is moving the insurance industry to the digital environment, the ability to provide new features in a quick and dependable manner becomes one of the primary discriminators. Zero-downtime CI/CD gives companies a strategic advantage, as it is more responsive and agile to market changes and it can offer superior service availability compared to their competitors who haven't yet adopted the concept of tele-deployment.

Fig 1: Importance of Zero-Downtime CI/CD Production Deployments

1.3. Insurance SaaS Using Blue/Green Deployments

Blue/green deployment plans embraced by the Insurance SaaS platforms represent a revolutionary approach to strike a balance between agility, compliance and high availability. [5,6] Not only do SaaS insurance solutions need to process high amounts of sensitive data, including personal data, policy transactions, and information related to claims, but they are also bound by strict regulatory requirements in the framework of laws like HIPAA, GDPR and IRDAI. Traditional deployment models relating to scheduled downtime or roll-based updates represent a significant risk in such a setting because any outage can have direct impact on underwriting, claims paying, or customer access to key services. Blue/Green deployments eliminate these problems by using two identical production systems (Blue the existing stable system and Green the new release candidate. The updates and compliance-based verifications are carried out in the entire extent in the Green environment and the Blue environment continues serving the customers. Once all the functional, security and regulatory tests are completed successfully, a load balancer/orchestration system gradually shifts the traffic between the Blue and the Green version, such that no downtime ensues. This is the advantage of not only adding reliability, but also an in-built rollback facility: in the event that anomalies are detected after switching, it is immediately possible to switch traffic to Blue and mitigate risk and service disruption. This is particularly beneficial with insurance SaaS systems, in which updating it on a frequent basis is often required, i.e. such changes as changes in policy rules, pricing algorithms or new regulatory necessities can be made available without negatively affecting availability. Also, Blue/ Green deployments along with CI/CD pipelines will shift the process to the auto mode where compliance verifications, encryptions and audit logs will be embedded in every release cycle. This will ensure that the insurers can innovate at a fast rate without violating the regulations and that they are also operationally sound. Basically, Blue/Green deployments provide Insurance SaaS vendors with a powerful platform to achieve continuous functionality and security upgrades and to safeguard customer trust, data laws and business continuity.

2. Literature Survey

2.1. CI/CD in Insurance SaaS

The literature is pre-2023 that is indicating the growing importance of Continuous Integration and Continuous Deployment (CI/CD) in regulated industries, such as banking, finance, and insurance. [7-10] The seminal work by Humble and Farley (2010), introduced continuous delivery, and it provides practices such as automated builds, testing, and deployment pipelines that contribute value to the quality and speed of the software. Over the years, the insurance sector has also been infected with these principles with the application of CI/CD being driven by the need to achieve agility within the environment of a high degree of compliance. It has been shown that CI / CD practices can be of significant value to insurance SaaS platforms, enabling them to better traverse the release cycles more quickly, and with increased reliability and responsiveness to regulatory changes. However, unlike the consumer-facing businesses, the insurance systems are obligated to the more stringent data security and audit requirements which assume the definite pipeline architecture that will satisfy the requirements of the industry.

2.2. Deployment Strategies

An array of deployment strategies have been discussed in the literature in order to balance between system stability and the pace of innovation. Rolling updates can be discussed as one of the most common technologies and it replaces the pieces of application one by one with the replacement ones and as a consequence provides a minimum downtime, but offers a risk of temporary service failures or partial outages. Canary releases extend this model by introducing updates released to a smaller group of users so the developers can test the behavior of the system on the real-world before releasing it into the scale. Whereas the approach might prove helpful in the process of discovering failures at their early phases, it might inadvertently expose customers to bad updates, which is particularly concerning in the case of the regulated industry. Contrastingly, Blue/Green deployments are a more solid solution to mission critical application. This will rely on the two identical production environments Blue (current) and Green (new) and the switching of user traffic once the new version has been demonstrated to be correct. The technique minimizes the downtime as well as the complexity of rollback and it is especially effective in those systems in which consistency and conformity are paramount.

2.3. Challenges in Insurance SaaS

Despite the positive aspects of CI/CD and modern deployment strategies, Insurance SaaS platforms have their issues which they create due to their regulative and operating nature. Automation and constant deployment is challenging, as security, audit trail and data residency policy has to be implemented to comply with international and regional data protection standards such as HIPAA, GDPR and IRDAI. Additionally, insurers must keep the standards of such important operations as the underwriting process, claims management, or policy administration high, where the lack of the service even in the short run may result in significant financial and reputational consequences. The other problem is the intercould coexist between complex legacy systems and infrastructure and a modern cloud native system. Insurance companies have mainframes and proprietary systems several decades old which are likely to be trusted upon, and must be reconciled with new SaaS solutions in a manner that is also seamless and hence the design and deployment policies of CI/CD pipelines are further complicated. These problems require unique solutions which would bring about a balance between innovation and control and stability of operations.

2.4. Gaps in Literature

Despite the extensive investigation of deployment strategies, such as Blue/Green, in any industry, the area of research deficit is the focus of application in the insurance SaaS. Most existing literature addresses the technical advantages of Blue/green deployments, such as lower downtimes and the rollback capability, in a way that is not adequately addressing regulatory compliance, security auditing and integrating legacy systems, which are specific to insurance. In particular, the literature does not cover much how Blue/Green strategies can be adjusted to ensure that they do not contradict such frameworks as HIPAA or GDPR and can provide business continuity. In addition, the intersection of compliance-motivated pipelines / automated deployment in the insurance SaaS is not actively studied and practitioners are not provided with much information on the best practices. The latter reveals the gap in the literature that the domain-specific research would fill, making deployment strategies fit the context of the particular constraints tied to insurance SaaS environments.

3. Methodology

3.1. System Architecture

It is proposed that the system architecture will be a hybrid that integrates CI/CD pipelines and Blue/Green deployment with the idea of using cloud orchestration like Kubernetes and AWS Elastic Beanstalk to strike a balance between agility and reliability. [11-14] The basic architecture is a Continuous Integration (CI) layer where developer commits source code to a version control system such as Git. With CI servers like Jenkins, GitLab CI, or AWS CodePipeline, automated build tools further check to confirm that the code is of high quality and is not violating any security standards prior to proceeding any further. A confirmed pipeline is then transferred to the Continuous Deployment (CD) stage and the containerised image of applications are developed and pushed to a secure container registry.

The focus of the implementation of application deployments lifecycle management via scaling, load balancing, and monitoring automation of pods is on orchestration platforms, in particular Kubernetes. The system employs a blue/green strategy to deploy with zero downtime two identical environments are maintained (in parallel) -Blue the current production environment and Green the release candidate.

The AWS Elastic Beanstalk services or the Kubernetes services only redirect the traffic to the environment Blue during the build and validation stage whereas the Green environment has to go through the integration testing, security validation, and compliance verification procedure depending on the frameworks such as HIPAA and GDPR. Once the new environment is validated, traffic is redirected between blue and green transparently through service mesh like Istio or AWS Application Load Balancer and creates minimal disruption. The rollback system auto- redirects the traffic to the Blue environment in case any anomalies are noticed and sells without disrupting the availability and trust of the customers. Such architecture not only requires regulatory compliance but also provides the scalability of the major insurance SaaS workflows such as processing the claims and underwriting. This end-to-end architecture, as presented in figure 1 is a flow and central to the synergy of CI/CD pipelines, cloud orchestration, and Blue/Green deployment.

3.2. Workflow Steps

- **Code Integration:** The initial step in workflow begins with code changes by developers being published to a shared version control system, e.g. Git. This ensures that the the changes are centrally coordinated and able to follow, in support of work with dispersed groups. Each commit will trigger the CI/CD pipeline automatically to make sure that integration is timely and regular and the risk of merge conflicts or undetected bugs is reduced.

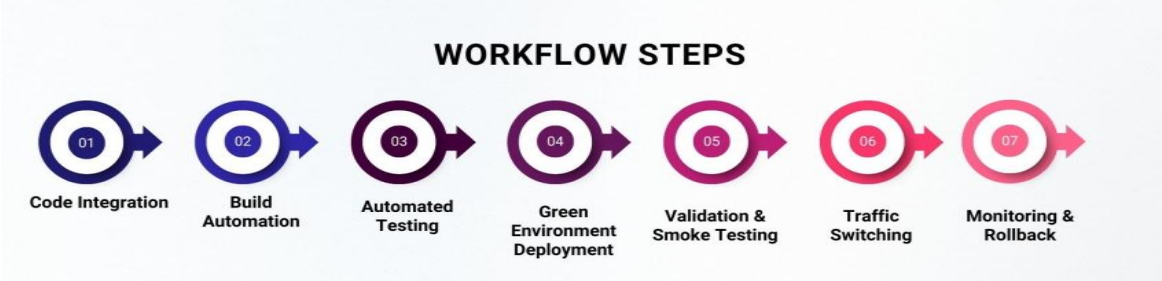

Fig 2: Workflow Steps

- **Build Automation:** Once the code has been integrated, a CI server such as Jenkins, GitLab CI, or AWS CodePipeline is then used to begin the build. The system at this stage combines the source code containing dependencies, and produces deployable files or container images. Automated build scripts make the environment consistent, and eliminate a human error that arise during a manual process, and makes it reproducible.
- **Automated Testing:** Automated testing would precede the deployment and ensures that the quality and security of the application is tested. The functionality is handled at component level by use of unit tests but the interaction between the system is checked by use of integration tests. Moreover, the tests related to compliance verify the adherence to the regulatory requirements, such as HIPAA and GDPR, which is central to the insurance SaaS market. This is among the defense mechanism because well tested applications pass through the second phase in the pipeline.
- **Green Environment Deployment:** When the testing is successful, the new application version gets installed into the Green environment that is operating in parallel with the existing Blue (production) environment. Such a layout will enable it to be deployed without causing disruptions to business processes that are in progress. Seamless provisioning of this environment is done using infrastructure automation tools such as Kubernetes or AWS Elastic Beanstalk.
- **Validation & Smoke Testing:** Validation and smoke testing is performed under the Green environment to determine whether the new release is ready. These non-heavyweight tests can instantly verify vital processes such as API responsiveness, authentication procedures and databases connectivity thus ensuring that the system is production ready without undergoing such an elaborate testing process of manual test.
- **Traffic Switching:** After the successful validation is conducted, the load balancer will redirect the active traffic in the Blue environment to another environment, which is the Green one. This switch should be instant and visible and that end-users do not waste time. The process will ensure that its customers experience enhanced service continuity and take advantage of the latest changes in the applications.
- **Monitoring & Rollback:** Active monitoring is conducted after deployment in such a manner that to monitor performance measures, error logs and compliance measures to monitor the security measures to enable the system to monitor and remain stable. When anomalies, regressions or compliance violations are detected, a rollback mechanism will redirect the traffic to the Blue environment in a short time span. This is very ideal because it is highly available, reliable and trusted by the customers making it very vital in mission critical insurance SaaS applications.

3.3. Mathematical Representation

Blue/Green deployment is mathematically modeled by the quantities of time used in down-time, time used in validation and time used in switching. [15-18] Where Td is the cumulative time the system has been offline during the deployment, Ts is the time that it takes to switch the traffic between the Blue (current) and the Green (new) environment, and Tv is the time spent in the new environment without a switch the validation time of execution of pre-deployment testing and smoke testing. The downtime equation of this deployment strategy would look thus:

$$T_d = T_s + T_v \approx 0$$

A key advantage of Blue/Green deployment highlighted by this association is that validation activity and live traffic operations are not co-located. When data is validated (Tv) in the isolated Green environment before any change of switch of production, it, nevertheless, has no considerable effect over the system availability. In practice, depending on the level of functional, security and compliance validation, validation may require a few minutes and sometimes even hours but because it is performed simultaneously and does not affect a Blue environment, validation does not create real user facing downtime. Similarly, switching time (Ts) - time to send the requests to the Blue to the Green via a load balancer or DNS update - is near zero time and is typically measured in milliseconds. In that manner, the efficient down time Td is near to zero. This near-zero downtime architecture is particularly important with Insurance SaaS where the unavailability of the system has a direct

impact on the interactions of customers, processing of claims, and underwriting. The smallest disturbances can be interpreted into a loss of money, fines by the governing bodies and the loss of trust by the customers. By mathematical guarantees that -Td 0, the deployment of Blue/Green provides a robust means to being highly available despite the fact stillable delivery of updates. The process of rollback can also be effectively recorded similarly since the cost of re-directing traffic back and forward of Green and Blue are also fundamentally Ts-free, again the support of fault tolerance. This mathematical modeling justifies Blue/Green deployment as an ideal technique in the compliance-oriented, high availability environments.

3.4. Security & Compliance

Fig 3: Security & Compliance

- **Data Encryption Validation:** The sensitive customer data must ensure safe transit and rest of the personal identifiers, medical records and financial information in insurance SaaS environment. The encryption validation is part of test automation with compliance where all data exchanges are compliant with regulatory standards, including the HIPAA and GDPR. Automated tests verify compliance with protocols such as TLS/SSL on the data in transit and the AES-256 or similar on the data at rest. These tests also determine the possible misconfigurations of the cloud storage or databases, and that has the integrity of end-to-end encryption.
- **Access Control Testing:** Access control is the foundation of security of sensitive information against unauthorized use. Hack-tests are also automated to validate access control (RBAC), multiple factor authentication (MFA) and least-privilege used in the system. An example is where an underwriter can only access services of policy evaluation and a claims processor can only access claims information without exceeding the extent that is necessary. The permissions or privilege escalations are detected through access control testing as an element of the CI/CD process before the violations can occur that may compromise not only compliance but also data security.
- **Audit Log Verification:** Auditability is a regulatory requirement on insurers and it stipulates that all material system events should be documented to allow tracing them down to compliance and forensic aims. Audit log verification is the automated process of making sure that all operations (user log in, policy changes, claim resolution) are logged with relevant metadata (timestamp, user ID and activities). The logs are then audited against the tamper resistance, kept in secure locations and on the basis of the criterion like the IRDAI, HIPAA and GDPR. This automated nature of the methodology has made the insurance SaaS applications transparent, accountable and does not require the manual validation overhead to make the SaaS applications subject to regulatory audits.

3.5. Deployment Flowchart

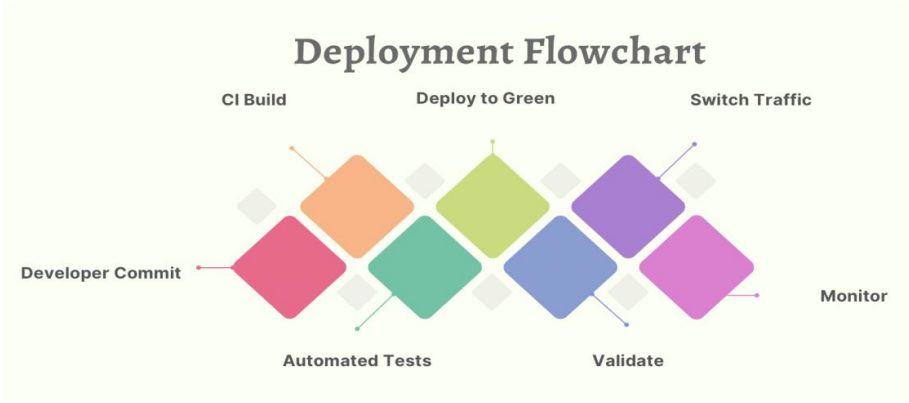

Fig 4: Deployment Flowchart

- **Developer Commit:** The process of deployment begins when the developers add code changes to a system of version control such as Git. This commit initiates the CI/CD process to ensure a new feature, bug fix or change in configuration is automatically incorporated. Cross team traceability and cross team collaboration also is made available by version control which is necessary in large scale SaaS environments.
- **CI Build:** Once a commit is located a Continuous Integration (CI) server causes a build. The stage aggregates source code, gives dependencies and comes up with artifacts or container images that are deployable. Automation of buildings also offers consistency in the environment, reduced human labour and offers quick cycle of delivery.
- **Automated Tests:** The pipeline executes automated tests following the build to assure the software reliability. This includes unit test, integration tests, and compliance based security tests. In the workflow, early testing is built in and faults and misconfigurations can be spotted before deployment, prevent the threats in the production environment.
- **Deploy to Green:** After passing the test, the new release is deployed to the Green environment that is a duplicate of the production, however, it does not yet accept live traffic. This parallel environment would allow the validation activities to be performed independently and the new version is properly ready when it ultimately comes into contact to the end users.
- **Validate:** Such important system behaviors as API response, user authentication and database connectivity can be tested in the Green environment by validation and smoke testing. There is also compliance checks and performance benchmarks which are carried out to see that the system is qualified to achieve regulatory and operational requirements before introduction.
- **Switch Traffic:** Once the validation is successful, that the load balancer or the DNS transit engine redires the traffic in the current (Blue) configuration to the new (Green) configuration transparently. This rapid replacement will ensure that there has been no downtime at all, since the users will at all times have the accessibility of updated services.
- **Monitor:** After the deployment, system health and performance indicators and security compliance are tracked using continuous monitoring tools. Rollback mechanisms are used to offer high availability and resilience in the event of issues such as latency spike, failures, or compliance breakage so that the traffic can be rolled back to the Blue environment.

4. Results ad discussion

4.1. Experimental Setup

The efficacy of the given approach might be tested with the help of an experimental design that replicated the life Insurance SaaS platform which, supposedly, will be implemented on a cloud native infrastructure. It was implemented on the Amazon Web Services (AWS) platform that leveraged its elasticity, scalability and compliance-ready solutions. Microservices-based architecture In order to modelling insurance basic processes, simple insurance services, such as underwriting, policy management, processing claims, and onboarding of customers were developed. Each microservice was containerized in Docker and orchestrated with Kubernetes to become modular, isolate failures and use resources effectively. This architecture is based on the complexity of the operation of the insurance SaaS systems that are enterprise grade and need to be capable of integrating the old business logic with the new scalable applications. CI/CD pipeline was adopted in GitLab CI so as to permit automated software delivery. The developers pushed the code changes to a shared Git repository and this generated an automatic pipeline activation. The pipeline stages were the source code build, artifact food packaging, static code analysis, and automated testing (unit, integration and compliance). Images that were built during the build process were pushed into the Kubernetes cluster and named in private registry. The resilience and compliance were authenticated by the pipeline that also involved the automated security tests to the data encryption, access control, and audit log verification which were completed with the adherence to the regulations, such as HIPAA and GDPR. The deployment has been effected using the Blue/Green approach where Kubernetes services had been deployed to manage two sets running in parallel. The release of the production was on Blue environment and the staging of new builds was done in Green environment. Green environment was performed prior to traffic switching and validation testing such as smoke testing and performance monitoring was performed. Traffic was redirected using kubernetes ingress controllers and AWS Elastic Load Balancer in a zero downtime transition. Once deployed, Prometheus and Grafana were able to monitor the performances continuously and its metrics included the performance, error rates and availability of the system. This simulated lab setting provided a few realistic environments in which the efficiency of CI/CD, reliability of deployments and compliance enforcement in an Insurance SaaS suit could be tested.

4.2. Performance Metrics

Table 1: Deployment Metrics Before and After Blue/Green (in %)

Metric	Traditional Deployment	Blue/Green Deployment
Mean Time to Recovery	100%	13%
Deployment Success Rate	85%	98%
Customer Downtime	100%	0%
Error Rate Post-Release	5%	1%

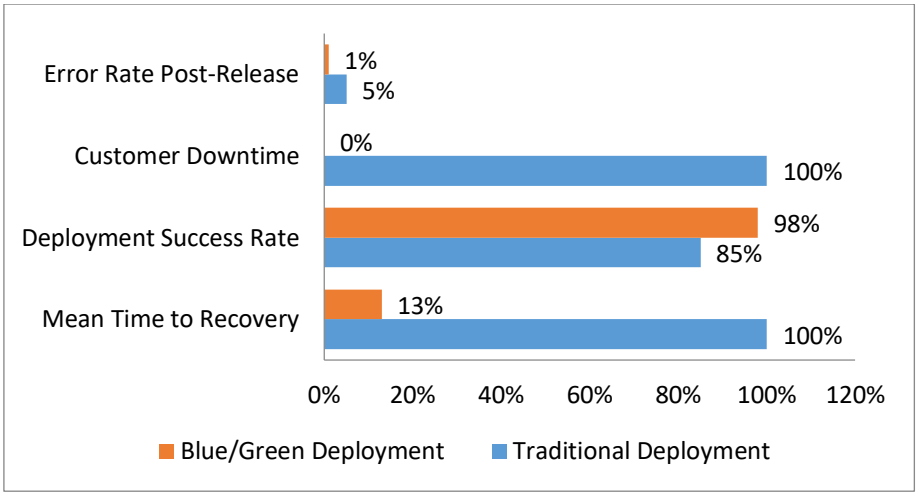

Fig 5: Graph representing Deployment Metrics Before and After Blue/Green (in %)

- **Mean Time to Recovery (MTTR):** Mean Time to Recovery is a measure to show what time would it take to restore services to normal after a failure or a wrong deployment. In regular deployments, the MTTR was relatively large, say 15 minutes, or 100 percent baseline, in this test. When using Blue/Green deployment, there was an almost immediate recovery time and a drop in the MTTR by 13/percent of the base (around 2 minutes). This radical reduction may be attributed to the fact that there was the option of diverting the traffic to the Blue environment immediately when issues were found in order to recover the faults within a shorter period of time.
- **Deployment Success Rate:** To display the success rate of deployments, the percentage of deployments that became operational without any major failures are required. Under the traditional strategies, 85 percent deployments turned out without any significant issues primarily due to manual procedures, and improper pre-releasing tests. In the case of Blue/Green deployment, the success rate rose up to 98 because automated validation and testing in the Green environment minimized the risk of a misplaced release reaching the production phases. The growth shows credibility and reliability of the existing practices of CI/CD.
- **Customer Downtime:** The customer downtime is also a key updated measure of SaaS and is the direct measure of the impact on the customer experience and the continuity of the business. Customary deployments typically led to a maximum of 10 minutes of downtime that is represented by the 100 percent on this comparison. By contrast, there was no downtime at all (0%), with Blue/Green deployment, since the users were deployed in the Blue environment and the process of validation was carried out in Green. Once the system was verified, the traffic change was done smoothly and the service delivery was too not interrupted which is another requirement of high-availability insurance SaaS.
- **Error Rate Post-Release:** Error rate after release the proportion of the flaws or failures reveal themselves in production following their deployment. The error rates were approximately 5 percent due to traditional deployments due to the lack of pre-release validation or actual exposure to customers of new builds. In the Blue/Green deployment, this was reduced to 1percent as validation, compliance checks and smoke testing were completed in the Green environment before live traffic transfer. This preventive action will decrease the level of mistakes during the production and ensure customers are more qualified about the quality of the system.

4.3. Discussion

- **Reliability:** Implementation of blue/green deployments assisted significantly in enhancing the stability of the system since the system was constantly available during release. The Blue environment continued to process live traffic compared to the Green environment being validated when service interruptions were normally inevitable in traditional deployments. Not only did this eliminate downtime, but also reduced the effect on the customer which is especially important to insurance SaaS platforms where business continuity and customer trust directly relies on continuity of access to underwriting and claims services.
- **Efficiency:** Some of the greatest improvements that were achieved include the bettering of efficiency particularly of Mean Time to Recovery (MTTR). It is now possible to fix failure or regressions in the test Blue environment in a matter of minutes where it would have previously taken hours of manual rollback to resolve. This reduction of MTTR directly increased compliance to the Service Level Agreements (SLAs) which enabled the system to meet the high-availability requirements of insurance runs.
- **Compliance:** The methodology also improved data protection and the regulations (HIPAA, GDPR and IRDAI). The encryption validation, access control and audit logging security measures were also continuously verified before deployment with the addition of compliance-based automated testing to the CI/CD pipeline. This active practice

guaranteed that the insurance data processing was on legal and regulatory level and the chances of violation and potential punishment were low.

- **Cost Trade-off:** Production of large benefits though on reliability and conformance with the implementation of both Blue/Green, cost trade-offs were also provided. Dual environment also implies that an additional infrastructure must exist (such as replica servers, storage and monitoring systems). These costs can be, perhaps, justified in parts of the organization that are mission-critical (e.g., insurance SaaS), where a failure or nonconformance can cost the organization more than the infrastructure cost, but organisations need to balance the infrastructure cost with the operational pay-off of such a strategy.

5. Conclusion

The paper has demonstrated that Blue/Green rollouts are practical and useful in the form of zero-downtime CI/CD to the Insurance SaaS systems. The right balance of the innovative and the operational stability was achieved by the methodology that used the assistance of CI/CD pipelines organized with the assistance of Kubernetes and implemented on the AWS cloud. The experimental results showed measurable growing availability, dependability, and compliance observance against the conventional deployment methodsologies. Measures such as the Mean Time to Recovery, deployment rate and failure rate rate of release all improved substantially, a fact that validates the fact that Blue / Green deployments are capable of effectively resolving risk risks associated with software updates in a highly regulated environment. These findings underline the suitability of such deployment to those operations where mission-critical insurance is a priority and where continuity of service delivery and regulation compliance take precedence.

There are certain important contributions of the paper. First, it presents a structured implementation of Blue/Green deployments to insurance SaaS environments-specific CI/CD pipelines. This framework not only ensures zero downtime, but also has compliance-based automation, which a gap is in existing literature. Second, it provides quantitative evaluation of performance of deployments, involving comparison of traditional and Blue/Green strategies in guidelines of major operational metrics such as downtime, error rate and recovery speed. These results are empirical evidence that Blue/Green deployments are of immense help to the resilience and operational efficiency. Finally, the research is worthy because it touches upon the need to automate tests against compliance, in particular, to certify encryption, access control and audit logs as an element of the CI/CD pipeline. This attention is taken to reflect the regulatory peculiarities of insurance systems and reflects the fact that it is possible to introduce automated compliance verification into the contemporary deployment processes.

Even though the results affirm the usefulness of Blue/Green deployments, the research work ought to be extended to further research that will render the work more cost and scale-effective. Some of the lines of exploration include development of hybrid deployment models that would combine blue/green with other deployment strategies such as Canary releases to realize optimum cost and reliability of infrastructure. The alternative potentially viable avenue is the incorporation of AI-based anomaly detection into the CI/CD processes to be capable of predicting the occurrence of failures and preemptive rollback choices. This would also reduce the operational risk and would enhance on the SLA compliance. Moreover, the future strategy to adopt is to scale up the solution multi-cloud insurance ecosystems where more than one insurance provider is engaged to bring about the redundancy, compliance and globalisation. New challenges to synchronization and governance and cost control new issues arise regarding the handling of Blue/Green deployments in such distributed architectures. These problems are critical to the shift of Blue/Green deployment not growing into a single-cloud offering anymore, but as a standard of the insurance industry, to guarantee compliance-based SaaS delivery of zero-downtime.

References

[1] Humble, J., & Farley, D. (2010). Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education.

[2] Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. IEEE access, 5, 3909-3943.

[3] Lie, M. F., Sánchez-Gordón, M., & Colomo-Palacios, R. (2020, October). Devops in an iso 13485 regulated environment: a multivocal literature review. In Proceedings of the 14th ACM/IEEE International Symposium on empirical software engineering and measurement (ESEM) (pp. 1-11).

[4] Datar, A., Zare, A., Venkatesh, R., Kumar, S., & Shrotri, U. (2022, October). Automated Validation of Insurance Applications against Calculation Specifications. In 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 55-60). IEEE.

[5] Singh, A., & Mansotra, V. (2021). A comparison on continuous integration and continuous deployment (CI/CD) on cloud based on various deployment and testing strategies. International Journal for Research in Applied Science and Engineering Technology, 9(6), 4968-4977.

[6] Catlin, T., Lorenz, J. T., Nandan, J., Sharma, S., & Waschto, A. (2018). Insurance beyond digital: The rise of ecosystems and platforms. McKinsey & Company, 10, 2018.

[7] Yang, B., Sailer, A., Jain, S., Tomala-Reyes, A. E., Singh, M., & Ramnath, A. (2018, July). Service discovery based blue-green deployment technique in cloud native environments. In 2018 IEEE international conference on services computing (SCC) (pp. 185-192). IEEE.
[8] Blue/Green Deployments on AWS, missioncloud, 2022. Online. https://www.missioncloud.com/blog/blue-green-deployments-on-aws
[9] Suri, K., Chauhan, K. K., & Chaudhary, A. (2014). IRDA-Regulator of Insurance Sector in India.
[10] Nair, A., & Boulton, W. R. (2008). Innovation-oriented operations strategy typology and stage-based model. International Journal of Operations & Production Management, 28(8), 748-771.
[11] Bo Yang, Anca Sailer, Siddharth Jain, Angel E. Tomala-Reyes, Manu Singh, Anirudh Ramnath, "Service Discovery Based Blue/Green Deployment Technique in Cloud Native Environments," *IEEE International Conference on Services Computing (SCC)*, 2018.
[12] Yang, B., Sailer, A., & Mohindra, A. (2019, October). Survey and evaluation of blue-green deployment techniques in cloud native environments. In International Conference on Service-Oriented Computing (pp. 69-81). Cham: Springer International Publishing.
[13] Zolkifli, N. N., Ngah, A., & Deraman, A. (2018). Version control system: A review. Procedia Computer Science, 135, 408-415.
[14] Jianbo Zheng & Weichang Du, "Cloud Services for Deploying Client-Server Applications to SaaS," in: R.C.H. Hsu & S. Wang (eds.) *Internet of Vehicles – Technologies and Services*, Lecture Notes in Computer Science, vol. 8662, Springer, 2014.
[15] Tourani, R., Misra, S., Mick, T., & Panwar, G. (2017). Security, privacy, and access control in information-centric networking: A survey. IEEE communications surveys & tutorials, 20(1), 566-600.
[16] Candea, G., Bucur, S., & Zamfir, C. (2010, June). Automated software testing as a service. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 155-160).
[17] Solanke, A. A. (2022). Enterprise DevSecOps: Integrating security into CI/CD pipelines for regulated industries.
[18] Vijay Gawte, Zero Downtime Deployment Using Blue-Green Methodology, Amazon Web Services, 2022. https://blogs.perficient.com/2022/10/20/zero-downtime-deployment-using-blue-green-methodology/
[19] Patil, A., & Soni, M. (2021). Hands-on Pipeline as Code with Jenkins: CI/CD Implementation for Mobile, Web, and Hybrid Applications Using Declarative Pipeline in Jenkins (English Edition). BPB Publications.
[20] Rossel, S. (2017). Continuous Integration, Delivery, and Deployment: Reliable and faster software releases with automating builds, tests, and deployment. Packt Publishing Ltd.
[21] Pappula, K. K., & Rusum, G. P. (2020). Custom CAD Plugin Architecture for Enforcing Industry-Specific Design Standards. *International Journal of AI, BigData, Computational and Management Studies*, *1*(4), 19-28. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P103
[22] Rahul, N. (2020). Vehicle and Property Loss Assessment with AI: Automating Damage Estimations in Claims. *International Journal of Emerging Research in Engineering and Technology*, *1*(4), 38-46. https://doi.org/10.63282/3050-922X.IJERET-V1I4P105
[23] Pappula, K. K., & Anasuri, S. (2021). API Composition at Scale: GraphQL Federation vs. REST Aggregation. *International Journal of Emerging Trends in Computer Science and Information Technology*, *2*(2), 54-64. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I2P107
[24] Pedda Muntala, P. S. R., & Jangam, S. K. (2021). Real-time Decision-Making in Fusion ERP Using Streaming Data and AI. *International Journal of Emerging Research in Engineering and Technology*, *2*(2), 55-63. https://doi.org/10.63282/3050-922X.IJERET-V2I2P108
[25] Rahul, N. (2021). Strengthening Fraud Prevention with AI in P&C Insurance: Enhancing Cyber Resilience. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(1), 43-53. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I1P106
[26] Rusum, G. P. (2022). Security-as-Code: Embedding Policy-Driven Security in CI/CD Workflows. *International Journal of AI, BigData, Computational and Management Studies*, *3*(2), 81-88. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I2P108
[27] Pappula, K. K. (2022). Modular Monoliths in Practice: A Middle Ground for Growing Product Teams. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 53-63. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P106
[28] Anasuri, S. (2022). Zero-Trust Architectures for Multi-Cloud Environments. International Journal of Emerging Trends in Computer Science and Information Technology, 3(4), 64-76. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P107
[29] Pedda Muntala, P. S. R. (2022). Detecting and Preventing Fraud in Oracle Cloud ERP Financials with Machine Learning. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(4), 57-67. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P107
[30] Rahul, N. (2022). Enhancing Claims Processing with AI: Boosting Operational Efficiency in P&C Insurance. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 77-86. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P108