International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

AI-Driven Deployment and Cloud Migration Strategies for Stable Software Releases

Santosh Kumar Nayak

IEEE, Senior Member, Member of IET

Abstract: ***The growing complexity of modern software systems has made manual deployment difficult to manage at scale. Artificial intelligence offers practical support across the deployment lifecycle, from traffic control to rollback decisions. This article discusses how AI supports blue and green deployment strategies through automated traffic orchestration, anomaly detection, predictive rollback, performance tuning, and user experience monitoring. It outlines the role of machine learning, deep learning, generative models, and reinforcement learning in improving release safety and system reliability. Common operational use cases such as load testing, circuit breakers, chaos engineering, caching, and cloud migration are also examined. The discussion shows how AI-enabled deployment reduces downtime, lowers release risk, and supports continuous improvement in production environments. The AI-Integrated Enhanced Deployment Strategy leverages machine learning and intelligent automation to revolutionize software delivery, optimizing CI/CD pipelines for greater speed, reliability, and accuracy. By automating testing, proactively detecting errors, and dynamically managing resources, this approach reduces deployment risks, overcomes infrastructure complexities, and ensures faster, more secure release cycles. This approach is critical for modern MLOps, allowing organizations to manage model drift and ensure AI models are consistently updated in production environments.***

Keywords: AI deployment, blue green strategy, cloud migration, automated rollback, system stability

1. Introduction

Organizations faced challenges of effectively managing and scaling manually deployment. With AI adoption, it becomes efficient from initial deployment to rollback mechanism.

1) **Automated traffic Orchestration:** AI algorithm analyze real time user traffic and system health to determine optimal pace and timing from shifting traffic from blue to green environment - **MLFlow or DVC**
2) **Anomaly detection:** ML Models monitor app matrices (latency error rates and CPU/Memory usage) and quantify deployment. Tools like - **Dadadog, Grafana, Open telemetry, prometheus**
3) **Predictive Rollback:** AI predicts potential failure based on historical data and trends, triggering automated rollback to stable environment - FluxCD, EvidentlyAI, NannyML, or Seldon Core
4) **Performance Optimization:** AI continuously compare **blue/green** environments recommending configuration changes to optimize performance and resource allocation
5) **User Experience Monitoring: Natural language** processing and sentiment analysis can access user feedback and support tickets to detect negative impact post-deployment. - **Active Learning**

Benefits:

- **Reduced Downtime:** AI enable faster detection and response to the issue and minimize service interruption.
- **Lower Risk:** Automated rollback and anomaly detection reduce the risk of faulty release
- **Continuous Improvement:** AI learns from each deployment and improve future release.

AI Model Types and considerations:

1) **Generative AI Models** - GPT Models, Stable Diffusion
   Best Use: Adopt Prompt Engineering Technique, Prompt Templates, Token limiting and prompt tuning libraries **LMQL and LangChain**
2) **Machine Learning Models:** Random Forest,XGBoost
   Best Use: ML Flow, Performance matrix, Feature Stores as Feast or Tecton
3) **Deep Learning Model**:**CNNs**(Convolutional Neural Networks), **RNNs** (Recurrent Neural Networks)
   Best Use: Safe point rollout, batch processing, GPU scheduling and **NVIDIA triton** inference server
4) **Reinforcement Learning Models:** Q- Learning, Deep Q-Network, Policy Gradient
   Best Use: Blue/Green deployment, Checkpoint to maintain model progress, **Ray RLib** to simplify large scale RL models

Use Cases to maintain stability:

- **Load Testing for Peak Traffic**: Locust, K6, Jmeter, BlazeMeter
- **Circuit Breaker Implementation**: Resilience4J or Envoy automatically halt the request
- **Chaos Engineering Resilience:** Gremlin, Chaos Mesh, LitmusChaos - Controlled failure management
- **Caching Strategies for latency improvement:** Radis, Cloudflare CDN, Vanish - in memory, contact at edge, http caching respectively used.

AI models for cloud migration strategies:

1) **Data sync:** Ensure data sync between present infra and cloud
   Tools: RCone, AWS Data Sync, AZure Data Factory,
2) **Hybrid Cloud Strategy:** Active workload in multiple env - active-active workload
   Tools: Anthos, Azure Arc, AWS outposts
3) **Migration Roll Strategy:** Active - Passive deployment Model Blue/Green
   Tools: Canary Deployment, Blue/Green deployment
4) **Data Security:** Ensure during migration data should encrypted in-transit
   Tools: Encrypt in-transit using TLS, at-rest using AWS KMS, Azure Key Vault or Google Cloud KMS

5) **AI Model Optimization specific to Cloud:** Cloud specific hardware accelerates
   Tools: TPUs in Google Cloud (vision model efficiency), AWS Inferential (Large scale inference), Azure NC-Series VMs (High performance AI models)

Suggested Cloud Tools for Migration:
1) **Azure Migrate:** Migration Planning
2) **AWS Application Migration Service:** Automating replication and failover
3) **Google Cloud Migrate:** Data Integrity

2. Conclusion

AI augments automate deployment strategy by automating monitoring, traffic management, and rollback decisions, resulting in safer and more efficient software releases.

Key Components and Impact:
- **Intelligent CI/CD Pipelines:** AI enables smarter, faster code integration, testing, and deployment, reducing manual intervention.
- **Predictive Operations:** Machine learning models predict and prevent potential deployment failures, enhancing system reliability.
- **Operational Efficiency:** AI optimizes resource allocation and manages containerized environments, reducing costs and improving performance.
- **Governance and Security:** AI automates compliance checks and mitigates bias in AI models, ensuring ethical deployment.
- **Modernization:** This strategy addresses legacy system limitations and streamlines the transition to containerized or cloud-native environments.

In this article, we have explores different AI models and use cases to make strategic production deployment for stable and robust approach.

References

[1] Azure Migrate - Step-by-Step Migration Planning: Azure Migrate Documentation
[2] AWS Application Migration Service - Replication and Failover: AWS Application Migration Service
[3] Google Cloud Migrate - Data Integrity Migration Tool: Google Cloud Migrate Documentation
[4] https://www.bain.com/how-we-help/whats-missing-from-your-ai-strategy-strategic-clarity/
[5] https://www.infotech.com/research/ss/build-a-scalable-ai-deployment-plan